

# **Efficient High-Order Accurate Methods using Unstructured Grids for Hydrodynamics and Acoustics**

**ONR Award Number N00014-04-0602**

**Final Report**

**August 31st, 2007**

**Prepared by**

**Professor Dimitri Mavriplis (PI)**

**Department of Mechanical Engineering**

**University of Wyoming**

**Laramie, WY 82071**

**Phone: (307)-766-2868**

**email: mavripl@uwyo.edu**

**and**

**Professor Dan Stanescu (co-PI)**

**Department of Mathematics**

**University of Wyoming**

**Laramie, WY 82071**

**Phone: (307)-766-2868**

**email: stanescu@uwyo.edu**

**20070907339**

## Summary of Project

The objective of this project has been the development of high-order accurate simulation techniques for fluid flow problems of interest to the US Navy, such as hydrodynamics and acoustics. Over the last decade there has been a growing appreciation of the inadequacy of current production level computational fluid dynamics codes in delivering sufficient accuracy for these types of problems as well as for resolving a sufficiently wide range of turbulence scales for reliable large eddy simulations. The use of higher-order discretizations such as Discontinuous Galerkin methods has been advocated as an alternate approach for simulating these types of problems with increased accuracy. While these methods have been under development for over two decades, most of the effort has concentrated on the development of sophisticated spatial discretizations, which in turn have been used mostly with explicit solution techniques. In order to be useful in an industrial setting, the efficiency of these methods must be improved through the development of more optimal solvers for steady-state and time-implicit problems.

In this project, we have concentrated on the development of efficient solution techniques for high-order Discontinuous Galerkin methods from both a theoretical and practical standpoint. The basic approach has been to develop and demonstrate an h-p multigrid solution strategy which delivers optimal convergence rates which are independent of both the order of accuracy  $p$  of the discretization, and the resolution  $h$  of the mesh. This solution technique has been demonstrated for both steady-state problems, and for time-implicit problems, using a high-order implicit Runge-Kutta time stepping scheme. The solution approach was originally implemented and demonstrated for the two-dimensional Euler equations, using spatial discretizations up to  $p=5$  (6th order accurate in space). Subsequently, these were extended to three-dimensional problems implemented in parallel, demonstrating good parallel efficiency on up to 2000 processors. Recently, the h-p multigrid solver has also been extended to the full Navier-Stokes equations in two dimensions, using the interior penalty (IP) method, and other approaches to viscous DG discretizations have been explored. While the initial multigrid approach has been performed in modal space, extensions of this solver to nodal DG discretizations have also been investigated.

At the same time, investigations into modern techniques for sensitivity analysis and error control using adjoint methods have been pursued. The discrete adjoint of the DG discretization has been derived and implemented, and demonstrated for a shape optimization problem. Solution of the adjoint problem is achieved using the h-p multigrid algorithm, yielding similar efficiency compared to the analysis problem. The adjoint solution has also been used to estimate spatial error and drive an adaptive h-p refinement procedure for improving accuracy in targeted output functionals at optimal cost.

The techniques developed in this project have demonstrated the potential of efficient high-order accurate methods such as the DG approach for industrial flow problems. Future work is underway, effectively leveraging the results of this project, in order to extend these techniques to the full three-dimensional Navier-Stokes equations running on massively parallel computer architectures.

This project, which was funded under the DEPSCoR program, has been instrumental in building the computationally oriented research programs of both the PI and co-PI on campus, and has fostered long-lasting collaborations between the University other federally funded research centers.



## Contents

<b>1. Introduction</b>	<b>1</b>
1.1 High-Order Discretizations . . . . .	2
1.2 Unstructured Mesh Techniques . . . . .	3
1.3 Solution Adaptive Approaches . . . . .	4
<b>2. Technical Achievements</b>	<b>5</b>
2.1 Governing Equations for Inviscid Flow Problems . . . . .	6
2.2 High-Order Discontinuous Galerkin Spatial Discretization . . . . .	6
2.3 The Implicit Solver . . . . .	8
2.4 The <i>hp</i> -Multigrid Approach . . . . .	9
2.5 Parallel Implementation . . . . .	10
2.6 Simple flow configuration . . . . .	11
2.6.1 Accuracy Validation . . . . .	12
2.6.2 Multigrid Efficiency Study . . . . .	13
2.7 Complex flow configuration . . . . .	14
2.7.1 Multigrid Results . . . . .	15
2.7.2 Computational Performance . . . . .	16
2.8 Extension to Implicit Time-Dependent Problems . . . . .	19
2.8.1 Vortex Convection Case . . . . .	21
2.8.2 Shedding flow over a triangular wedge . . . . .	25
2.9 Extension to Diffusion Problems and the Navier-Stokes equations . . . . .	28
2.9.1 Unified Inviscid-Viscous Flux Formulation . . . . .	28
2.9.2 Interior Penalty Method . . . . .	29
2.10 <i>p</i> -Multigrid for Nodal DG Approximation . . . . .	32
2.11 Adjoint-Based Error Estimation and Adaptivity . . . . .	34
<b>3. Conclusions and Further Work</b>	<b>38</b>
<b>4. Broader Impact of DEPSCoR Project</b>	<b>38</b>
4.1 People Supported by the Grant . . . . .	39
4.2 Publications Resulting from this Grant . . . . .	39

## 1. Introduction

Computational fluid dynamics (CFD) is now accepted as an integral part of most engineering design and analysis exercises. Within the DoD, CFD is used extensively for modeling new concepts, understanding physical phenomena, and aiding in the design process. In fact the use of CFD has become so ubiquitous, that it is often considered by many to be a mature engineering tool, and investment has often shifted from the development of better analysis capability, to the incorporation of existing simulation capabilities into more complex coupled multidisciplinary analyses, or design optimization strategies.

However, developers and practitioners alike fully understand the severe limitations of current-day CFD capabilities. The most well-known of these is probably the difficulties in the simulation of turbulent flows. Because of the wide range of scales present in high-Reynolds number turbulent flows of interest in engineering problems, it is not feasible to resolve all relevant scales in a numerical simulation, and physical modeling of the turbulence must be performed in the form of a Reynolds-averaged Navier-Stokes (RANS) approach. RANS approaches with turbulence models have achieved reliable predictive abilities only for the small subset of flows where separation is either non-existent, or of a minimal nature. For massively separated flows, the alternative of using Large Eddy Simulation (LES), which attempts to numerically resolve the most important range of turbulent scales, appears daunting because of the increased expense of such simulations, which is easily several orders of magnitude larger than a RANS simulation [1].

Other phenomena which have proved difficult to simulate numerically because of disparate length scales include wave phenomena, such as acoustics and electromagnetics, particularly as the wave frequency increases. Even steady-state mean flow simulations, such as the flow over a maneuvering submarine, or a missile, have proven very difficult to compute accurately, particularly if the effect of convected flow features such as vortices and wakes is of concern [2, 3]. The resolution of such flow features typically degrades very quickly as these progress downstream in the simulation, due to the numerical diffusion inherent in any computational approach. In fact, the simulation of such engineering flows still has yet to be computed sufficiently accurately and routinely for use in a design cycle.

In some sense, the state-of-the-art in CFD is currently at a cross-roads. This field has achieved a high level of utility in science and engineering, becoming an almost indispensable component of any design process. On the other-hand, the remaining deficiencies are so serious that CFD can only be used confidently in a relatively narrow application range. Furthermore, rapid advances in simulation capability have not been forthcoming over the last decade, apart from the increased affordability due to the decreasing cost of computer hardware. The majority of current-day CFD solvers rely on algorithms developed ten to twenty years ago, which have been refined and made more accessible through the evolution of software and the constant decreasing cost of hardware.

One major advance has been the increasing development and acceptance of unstructured grid methods. These methods have resulted in greater flexibility for the modeling of complex geometries, most often encountered in real-world engineering problems. One of the great promises of unstructured grid technology is the ability to easily incorporate adaptive meshing techniques. However, adaptive meshing techniques have seldom achieved their potential for engineering problems, due to the inherent difficulties in devising reliable error estimation criteria to guide the adaptive grid refinement process.

To make substantial progress in extending the scope and utility of CFD, there is an emerging consensus that a more sophisticated and comprehensive approach will be required, other than reliance on the usual well-established techniques developed over the past twenty years.



Several important recent developments are seen as having a potentially large impact on the future capability of CFD. One of these is the development of higher-order discretizations, and the tailoring of these approaches to unstructured mesh methods. Another area of interest involves recent advances in output functional error estimation. It is the further development of these ideas and incorporation into an unstructured grid framework which will enable substantial future advances in CFD capability.

### 1.1 High-Order Discretizations

While most traditional CFD discretizations are second-order accurate, higher-order accurate discretizations have been proposed and investigated from an early stage [4–6]. The basic idea can be surmised from simple approximation theory. If a continuous function  $f$  is to be approximated over a set of discrete intervals of width  $h$ , a second-order accurate approximation (or linear fit) results in two degrees of freedom over each interval, corresponding to the mean value and slope of the function over the interval. Using a Taylor series expansion, an approximation can be written as:

$$f(x + \varepsilon h) = f(x) + \varepsilon \frac{h}{1!} \frac{\partial f}{\partial x} + \varepsilon^2 \frac{h^2}{2!} \frac{\partial^2 f}{\partial x^2} + \varepsilon^3 \frac{h^3}{3!} \frac{\partial^3 f}{\partial x^3} + \varepsilon^4 \frac{h^4}{4!} \frac{\partial^4 f}{\partial x^4} + \dots \quad (1)$$

where  $0 < \varepsilon < 1$  for approximations inside the interval of width  $h$ . For a second-order approximation, which is exact for linear functions, the approximation error is given as:

$$Error = Ch^2 \frac{\partial^2 f(\eta)}{\partial x^2} \quad (2)$$

where  $\eta$  is in the interval  $\eta \in (x, x + h)$  and  $C$  is a constant. For higher-order approximations, for example fourth order-accurate cubic interpolations often used in spline curve fitting, the larger number of degrees of freedom per interval (four in the case of cubics) enables the approximation of higher derivatives in the function  $f$ , thus resulting in an approximation error given as:

$$Error = Ch^4 \frac{\partial^4 f(\eta)}{\partial x^4} \quad (3)$$

where  $\eta$  is in the interval  $\eta \in (x, x + h)$ . Thus, because  $h$  is a small quantity, the higher-order error term is generally smaller than the lower-order error term, provided the higher-order derivatives of the function  $f$  are bounded. Perhaps more important is the asymptotic behavior of these approximations with respect to the interval size  $h$ : the linear error term varies as  $O(h^2)$ , while the cubic approximation term varies as  $O(h^4)$ . The implications are such that a doubling of grid resolution (halving of size of  $h$ ) results in an error reduction factor of 4 for the second-order scheme, and an error reduction factor of 16 for the fourth-order scheme. For a three-dimensional problem, the doubling of grid resolution results in a factor  $2^3 = 8$  increase in overall number of grid points and hence total computational work. Therefore, for second-order accurate schemes, as the grid is refined, overall computational costs increase more rapidly than the delivered accuracy. However, the superior asymptotic properties of higher-order discretizations are not sufficient by themselves to justify their use. From the above discussion, the absolute value of the error is seen to depend on the values of higher-order derivatives of the approximated function. Therefore, higher-order methods will be best suited for functions with bounded or small high-order derivatives, i.e. smooth functions. For non-smooth functions, including functions which may contain discontinuous higher derivatives, lower-order discretizations may provide higher accuracy at equivalent resolution levels. For example, it is well known that discontinuous solutions such as shock waves are most efficiently represented using low-order discretizations. Even for continuous solutions, the sampling (Nyquist) theorem[7] states that a given solution feature (i.e. wave



length) requires a certain minimum level of sampling points in order to be resolved by any method of any order. For equivalent resolution levels, low-order methods are intrinsically more cost-effective approaches, because of the relative simplicity of their construction. On the other hand, as the resolution is increased, superior error reduction will be achieved by higher-order methods. Thus, the most cost effective method depends implicitly on the level of desired accuracy, with the cross-over point between methods being a function of the "smoothness" of the final solution, and the relative costs of each scheme.

For accuracy levels and solution characteristics relevant to most engineering fluid flow problems, the demonstration of cost-effective higher-order methods has generally been lacking, and second-order methods have remained the preferred choice. However, a brief look at other fields, such as computational elasticity, and computational geometry (i.e. CAD packages, surface/curve representations) reveals extensive use of higher-order discretizations as the most efficient approaches. The key to realizing the benefits of higher-order discretizations for engineering CFD problems lies in the development of efficient solution algorithms for these sophisticated discretizations, and a careful matching of the most appropriate discretization types to the character (smoothness) of the computed solution, through solution adaptive techniques.

## 1.2 Unstructured Mesh Techniques

The use of unstructured meshes has gained widespread acceptance over the last ten to fifteen years, largely due to the increased flexibility such methods offer for dealing with complex engineering geometries, as well as the development of cost-effective discretizations and solution algorithms for unstructured meshes. The differentiation between structured and unstructured meshes is based not on element types, but on the data-structures employed by these different approaches. Unstructured meshes are often constructed using tetrahedral elements in three dimensions, but can be formed using hexahedral elements, prismatic or pyramidal elements. Their main characteristic is that there is no direct mapping of these collections of elements to a three dimensional coordinate system, so that the connectivity of the mesh must be stored explicitly. The implications are that often-used structured mesh solution techniques, such as banded matrix approaches, line solvers, and multigrid methods, can not be used directly on unstructured mesh discretizations. This resulted in the early adoption of simple explicit solution techniques for unstructured meshes, leading to inefficient overall solution capabilities. Ultimately, it has been the development of more efficient solution techniques, based on sparse matrix technology [8–11] or multigrid methods [12–15] and line solution algorithms adapted to unstructured meshes through specialized data-structures [16], which have driven the adoption of unstructured meshes into the engineering CFD community.

In a similar fashion, the initial development of higher-order discretizations for CFD methods was geared towards structured meshes. Finite difference approximations based on Taylor series expansions of the approximated function provided the most obvious path towards the achievement of higher-order accuracy [4, 5, 17, 18]. Unfortunately, these methods proved difficult to extend to unstructured mesh topologies. More elaborate approaches such as reconstruction techniques were proposed for unstructured meshes [19], although these suffered from implementation and efficiency issues due to the complex stencils involved. The development of finite-element approaches such as the Streamwise Upwind Petrov Galerkin (SUPG) or the Streamline Diffusion method [20, 21] and the Discontinuous Galerkin (DG) method [22, 23] has shown more promise for the use of higher-order discretizations on unstructured meshes. The SUPG approach relies on the approximation of the solution through a set of basis functions which span individual elements, but which are continuous across inter-element boundaries, in the traditional Galerkin finite-element sense [24]. Such discretizations are stable for diffusion problems, but require additional dissipation for ensuring the



stability of convection problems. Thus an additional streamwise diffusion term is added, the form of which must be carefully crafted to avoid compromising the higher-order accuracy properties of the scheme.

The Discontinuous Galerkin method was originally formulated by Reed and Hill [25] for the discretization of the neutron transport equation. This method was later refined and popularized using a different approach introduced in a series of papers by Cockburn and Shu [22, 23, 26, 27]. The DG method employs element-based basis functions formed as higher-order polynomials, in the same fashion as SUPG schemes. However, continuity of the basis functions across element boundaries is not enforced. The resulting solution jumps across element boundaries are resolved using a Godunov or approximate Riemann flux function [28, 29]. This approach has the advantage that the solution, expressed as modal coefficients within each element, only requires the inversion of an element matrix, once element boundary fluxes are resolved. One can safely state that the recent interest in DG methods, and in particular in their application to hyperbolic problems, as for example the Euler equations of gas dynamics, is mainly due to two reasons. Firstly, a large experience base was already in place in the CFD community as regards the application of high-resolution finite-volume methods to various types of flows, and the DG method can be viewed as a generalization of these methods for non-constant test functions (with, however, the promise of a more systematic approach to building high-order solutions), so the extension was straightforward. Secondly, the DG method offers the promise of a higher-order truncation error in the case of hyperbolic equations, and Jiang and Shu [30] proved that the DG method satisfies a local entropy condition for arbitrary triangulations in any space dimension and to any order of accuracy, which also implies stability of the method for nonlinear shock problems in the scalar case.

While much work has been devoted in the past to developing high-order accurate discretizations, techniques for efficiently solving the equations arising from these discretizations, in the context of steady-state or implicit time-integration problems, have not received as much attention. This is important, for the engineering success or failure of these methods ultimately rests on their overall cost effectiveness as regards to second-order accurate methods. For the majority of engineering problems, including steady-state problems often encountered in design optimization exercises, and unsteady problems where the time scales are significantly larger than the resolved spatial scales, explicit methods can not be expected to provide a competitive solution strategy. The project has therefore focused on the development of efficient solution methods for unstructured mesh high-order discretizations, while using relatively well established high-order DG discretizations.

### 1.3 Solution Adaptive Approaches

One of the advantages of unstructured mesh strategies involves the possibility of incorporating adaptive meshing techniques to enhance solution accuracy. In the context of higher-order accurate discretizations, the relative merits of increasing the grid resolution must be weighed against those obtained by increasing the order of accuracy of the discretization. This leads to the notion of h-p refinement, where h-refinement refers to the spatial refinement of the mesh, while p-refinement denotes the attainment of increased solution accuracy through the use of a (locally) higher-order discretization. Effective implementation of h-p refinement strategies require not only an estimate of the local solution discretization error, but also a characterization of this error in terms of smoothness. For example, a purely h-refinement strategy can at best provide a constant factor in error reduction, as equidistribution of error is ultimately achieved, and further reductions in overall solution error require global grid refinement. On the other-hand, the effectiveness of a purely p-refinement strategy may be impeded by the presence of solution discontinuities or non-smooth features (on the level of the existing grid resolution) which are not efficiently approximated using higher-order meth-



ods. At any level, the refinement decision must rest on an estimate of the local error relative to the overall global error, and the choice of refinement type is given by the relative local smoothness of the solution, with smooth solutions regions benefiting most from p-refinement, and rough solutions regions benefiting most from h-refinement.

The application of combined h-p refinement has been shown to result in exponentially rapid error convergence, thus providing an optimal strategy for accuracy improvement [31, 32]. However, the estimation of local discretization error has proven to be a major impediment towards the successful use of solution adaptive techniques, even in the simple context of h-refinement alone. Virtually all local error estimates assume the solution is in the asymptotic convergence range, which is often not the case for non-linear problems.

Rather than attempting to quantify the discretization error at every point in the mesh for adaptive refinement criteria, an alternative approach involving adjoint-based error prediction methods has been gaining popularity recently [33–35]. In this approach, solution of the adjoint of the governing flow equations is used to quantify the sensitivity of an objective function with respect to local grid resolution throughout the domain. Adjoint sensitivity techniques have been well developed in the context of design-optimization problems [36–38]. This approach to mesh refinement is non-local, in the respect that it provides for the global effect of local resolution changes. Objective functions representing engineering quantities of interest such as lift, drag, or  $L/D$ , can be constructed, thus enabling the mesh refinement process to target quantities of interest. Since vastly different grid resolution distributions are required for different engineering applications of the same problem (i.e. drag prediction versus sonic boom prediction), this approach enables accurate and efficient engineering results by avoiding excessive resolution in areas of little influence on the quantity of interest.

In the current project, we investigate the use of adjoint-based error estimation techniques for driving adaptive h-p refinement strategies, an area which has remained relatively unexplored, but holds the opportunity for radical improvements in engineering simulation fidelity and efficiency.

## 2. Technical Achievements

The principal long term goal of this project is the development of an efficient high-order accurate CFD solver suitable for Naval applications such as hydrodynamics and acoustics. While this ultimately requires the development and validation of a three-dimensional Navier-Stokes solver, a multi-pronged strategy was pursued throughout the project. Firstly, an inviscid (Euler) solver was developed from scratch, initially in two-dimensions, and then in three dimensions. Proceeding directly to three dimensions with the simplified setting of inviscid flow allowed us to rapidly build the required infrastructure for large scale multigrid and parallel computing problems. The two-dimensional code was retained and used to validate new algorithmic techniques, such as viscous term formulation, and adjoint sensitivity and adaptivity methods. At the same time, more fundamental investigations into viscous flux formulations and nodal multigrid methods were pursued in separate software development efforts.

In the following subsections, we first describe the discretization of the Euler equations and the development and implementation of the h-p multigrid algorithm for steady-state problems in three dimensions, including parallel scalability results. We next describe the extension of the multigrid algorithm to time-implicit problems, which was carried out for the Euler equations in two-dimensions. Extensions of the current formulation to the Navier-Stokes equations in two-dimensions are then discussed. Investigations



into the application of the h-p multigrid approach for nodal DG formulations are then addressed. Finally, adjoint-based techniques for error estimation and adaptive h-p refinement are described and demonstrated.

## 2.1 Governing Equations for Inviscid Flow Problems

The conservative form of the compressible Euler equations describing the conservation of mass, momentum and total energy are given in vectorial form

$$\frac{\partial \mathbf{U}(\mathbf{x}, t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \quad (4)$$

subject to appropriate boundary and initial conditions within a three-dimensional domain  $\Omega$ . Explicitly, the state vector  $\mathbf{U}$  of the conservative variables and the Cartesian components of the inviscid flux  $\mathbf{F} = (\mathbf{F}^x, \mathbf{F}^y, \mathbf{F}^z)$  are:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{pmatrix}, \quad \mathbf{F}^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E_t + p) \end{pmatrix}, \quad \mathbf{F}^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E_t + p) \end{pmatrix}, \quad \mathbf{F}^z = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E_t + p) \end{pmatrix}, \quad (5)$$

where  $\rho$  is the fluid density,  $(u, v, w)$  are the fluid velocity Cartesian components,  $p$  is the pressure and  $E_t$  is the total energy. For an ideal gas, the equation of state relates the pressure to total energy by:

$$p = (\gamma - 1) \left[ E_t - \frac{1}{2} \rho (u^2 + v^2 + w^2) \right] \quad (6)$$

where  $\gamma = 1.4$  is the ratio of specific heats.

## 2.2 High-Order Discontinuous Galerkin Spatial Discretization

The computational domain  $\Omega$  is partitioned into an ensemble of non-overlapping elements and within each element the solution is approximated by a truncated polynomial expansion

$$\mathbf{U}(\mathbf{x}, t) \approx \mathbf{U}_p(\mathbf{x}, t) = \sum_{j=1}^M \mathbf{u}_j(t) \phi_j(\mathbf{x}) \quad (7)$$

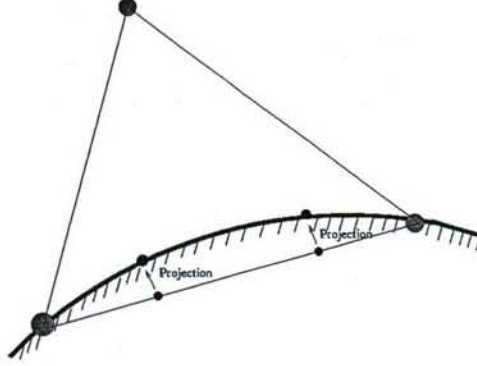
where  $M$  is the number of modes defining the truncation level. The semi-discrete formulation (*i.e.* continuous in time) employs a local discontinuous Galerkin formulation [39–42] in spatial variables within each element  $\Omega_k$ . The weak formulation for Eq. (4) is obtained by minimizing the residual with respect to the expansion function in an integral sense:

$$\int_{\Omega_k} \phi_i \left[ \frac{\partial \mathbf{U}_p(\mathbf{x}, t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}_p) \right] d\Omega_k = 0 \quad (8)$$

After integrating by parts the weak statement of the problem becomes:

$$\int_{\Omega_k} \phi_i \frac{\partial \mathbf{U}_p}{\partial t} d\Omega_k - \int_{\Omega_k} \nabla \phi_i \cdot \mathbf{F}(\mathbf{U}_p) d\Omega_k + \int_{\partial\Omega_k} \phi_i \mathbf{F}^*(\mathbf{U}_p) \cdot \mathbf{n} d(\partial\Omega_k) = 0 \quad (9)$$

The local discontinuous Galerkin approach makes use of element-based basis functions, which results in solution approximations which are local, discontinuous, and doubled valued on each elemental interface. Monotone numerical fluxes are used to resolve the discontinuity, providing the means of communication



**Figure 1:** Two-dimensional illustration of projection of additional surface points for creation of curved surface element.

between adjacent elements and specification of the boundary conditions. This local character of the discontinuous Galerkin method simplifies tremendously the parallel implementation, as will be shown later. The numerical flux,  $\mathbf{F}^*(\mathbf{U}_p) \cdot \mathbf{n}$ , is obtained as a solution of a local one-dimensional Riemann problem and depends on the internal interface state,  $\mathbf{U}_p^-$ , the adjacent element interface state,  $\mathbf{U}_p^+$  and the orientation as defined by the normal vector,  $\mathbf{n}$ , of the interface. An approximate Riemann solver is used to compute the flux at inter-element boundaries. Current implementations include the flux difference splitting schemes of Rusanov [43], Roe [44], HLL [45] and HLLC [46–48].

The discrete form of the local discontinuous Galerkin formulation is defined by the particular choice of the set of basis functions,  $\{\phi_i, i = 1 \dots M\}$ . The basis set is defined on the master element  $\hat{\Omega}(\xi_j, j = 1 \dots 3)$  spanning between  $\{-1 < \xi_j < 1\}$ . We seek a set of hierarchical basis functions in order to simplify our subsequent spectral multigrid implementation. The basis set contains *vertex*, *edge* and *bubble* functions [49, 50] based on Jacobi polynomials of variable weights. Since the basis set is defined in the master element, a coordinate transformation,  $\mathbf{x}_p = \mathbf{x}_p(\xi_1, \xi_2, \xi_3)$ , is required to compute the derivatives and the integrals in physical space  $\Omega_k(x, y, z)$ . For iso-parametric elements, the basis functions are expressed as functions of  $\xi_1$ ,  $\xi_2$  and  $\xi_3$ , and the coordinate transformation, and its Jacobian are given by:

$$\mathbf{x}_p = \sum_{j=1}^M \hat{\mathbf{x}}_j \phi_j(\xi_1, \xi_2, \xi_3), \quad J_k(\xi_1, \xi_2, \xi_3) = \left| \frac{\partial(x, y, z)}{\partial(\xi_1, \xi_2, \xi_3)} \right| \quad (10)$$

In the simple case of straight-sided or -faced elements the mapping is linear and its Jacobian,  $J_k$ , and its metrics are constant within each element, and can be evaluated just by using the element vertex coordinates. In the case of elements with curved faces, the Jacobian  $J_k$  varies within the element, and must be evaluated explicitly at the individual quadrature points in the numerical integration process. While straight-sided/faced elements are used in the interior of the domain, curved boundary elements which conform to the original description of the boundary geometry must be used in order to retain the  $p + 1$  accuracy property of the discretization scheme. Since most mesh generation packages produce a list of elements, with coordinates given only for the corner points of the elements, additional information is required in order to create curved boundary elements. This is achieved by projecting additionally created surface points onto the original boundary geometry description, as depicted in Figure 1 for the two-dimensional case.

The number of additional surface points to be created on each face is a function of the  $p$ -order of the discretization. These points are initially created on the flat face of the non-curved geometry using linear interpolation, and then projected onto the surface of the original defining geometry used in the grid



generation process. Throughout this project, the VGRID grid generation program [51] is used to generate three-dimensional unstructured tetrahedral meshes. A projection utility which snaps an arbitrary list of points to the closest surface geometry used in VGRID was developed and provided for this purpose [52]. The correspondence between the coordinates of the new projected surface points in physical space and in isoparametric space is then used to compute the exact Jacobian for the curved elements.

While elements with a face on the geometry boundary must be treated as curved elements, additional neighboring elements are also required to be treated as curved elements for complete consistency. A simple algorithm for determining the extent of the set of curved elements begins by identifying all elements with one or more faces on the boundary. This constitutes the set of elements where all faces must be considered curved. The second step is to propagate the cell coefficients back to all elemental faces, which might have only one curved edge on the curved surface. The third step is to transfer all face coefficients to all adjacent elements, even if they are not physically on the curved surface. This will guarantee that all the curvature information is exact and matched for all elements close to the curved surface.

For the general case (*i.e.* curved elements), using Eq. (10), the solution expansion and the weak statement within each element,  $\widehat{\Omega}_k$ , becomes:

$$\mathbf{U}_p(\xi, \eta, t) = \sum_{j=1}^M \widehat{\mathbf{u}}_j(t) \phi_j(\xi_1, \xi_2, \xi_3) \quad (11)$$

$$\int_{\widehat{\Omega}_k} \phi_i \frac{\partial \mathbf{U}_p}{\partial t} |J_k| d\widehat{\Omega}_k - \int_{\widehat{\Omega}_k} \nabla \phi_i J_k^{-1} \cdot \mathbf{F}(\mathbf{U}_p) |J_k| d\widehat{\Omega}_k + \int_{\partial \widehat{\Omega}_k} \phi_i \mathbf{F}^*(\mathbf{U}_p) \cdot \mathbf{n} |J_k| (d\widehat{\Omega}_k) = 0 \quad (12)$$

The resulting semi-discrete form, Eq. (12), can be further simplified as:

$$\mathbf{M} \frac{d\mathbf{U}_p}{dt} + \mathbf{R}(\mathbf{U}_p) = 0 \quad (13)$$

where  $\mathbf{M}$  and  $\mathbf{R}(\mathbf{U}_p)$  represent the mass matrix and the non-linear residual associated with the master element  $\widehat{\Omega}_k$ , respectively. This system of ordinary equations, Eq. (13), is solved in the modal space and the integrals are evaluated by economical Gaussian quadrature rules [49, 53, 54], which requires a projection of the solution values to the quadrature points used in the numerical integration. In order to preserve the  $p+1$  accuracy order of the numerical approximation, the element integral uses quadrature rules which are exact for polynomial degree  $2p$  within the master element, while the boundary integral uses quadrature rules which are exact for polynomial degree  $2p+1$  [55]. For boundary elements with curved edges or faces, the Jacobians must be evaluated at the integration quadrature points, whereas for interior elements with straight edges or faces, these are constant and need only be evaluated once for each element.

### 2.3 The Implicit Solver

For steady-state cases, the temporal derivative term may be omitted and the system of equations (Eq. (13)) associated with each element becomes:

$$\mathbf{R}(\mathbf{U}_p) = 0 \quad (14)$$

where  $\mathbf{R}(\mathbf{U}_p)$  now represents the non-linear steady-state residual. We use variants of an element-Jacobi scheme to solve this system of equations. The Newton iteration associated with Eq. (14) yields at each “ $n+1$ ” step:

$$\begin{aligned} \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{U}_p} \right]^n \Delta \mathbf{U}_p^{n+1} &= -\mathbf{R}(\mathbf{U}_p^n) \\ \mathbf{U}_p^{n+1} &= \mathbf{U}_p^n + \alpha \Delta \mathbf{U}_p^{n+1} \end{aligned} \quad (15)$$

where  $\alpha$  is a parameter used for robustness to keep  $\|\alpha \Delta \mathbf{U}_p^{n+1} / \mathbf{U}_p^{n+1}\|_{L_\infty} \leq 10\%$ . An element-Jacobi scheme can be defined as an approximate Newton scheme where the full Jacobian matrix is replaced by the block diagonal entries representing the coupling between all modes within each element,  $[\partial \mathbf{R} / \partial \mathbf{U}_p] \approx [D]_i$ , thus neglecting the coupling between neighboring element modes, which arises through the inter-element flux evaluations. The  $[D]$  blocks represent small dense matrices associated with each grid element. These element matrices are inverted using Gaussian elimination to produce a lower-upper (LU) factorization of each element matrix. In the case of the three-dimensional Euler equations (Eq. (4)) the number of entries in the block diagonal matrix ( $[D]$ ) for each tetrahedral element is given in Table (1). Clearly, the growth in size of  $[D]$  is a non-linear function of the discretization order, and dictates the memory requirement. Therefore, in our simulations we keep the discretization order to  $p \leq 6$ , which should be reasonable for hydrodynamic applications. The non-linear iteration Eq. (15) becomes:

$$\Delta \mathbf{U}_p^{n+1} = [D^n]^{-1} (-\mathbf{R}(\mathbf{U}_p^n)) \quad (16)$$

This solver is denoted as the non-linear element Jacobi (NEJ). A second variant of this solver is the *quasi* non-linear element Jacobi (qNJ). This variant employs “ $k$ ” quasi non-linear iterations, where only the residual,  $\mathbf{R}(\mathbf{U}_p^k)$ , is updated, and the block diagonal matrices,  $[D^n]$ , are kept constant throughout the outer-iteration “ $n$ ”. Therefore, the  $(k+1)^{th}$  step is:

$$\Delta \mathbf{U}_p^{k+1} = [D^n]^{-1} (-\mathbf{R}(\mathbf{U}_p^k)) \quad (17)$$

where the  $[D^n]^{-1}$  matrix is actually stored in LU factorized form. This approach is expected to yield similar converge rates per cycle as in the NEJ variant, with improved performance in terms of CPU time, since the expensive  $[D]$  matrix assembly and LU factorization procedure are performed less frequently. In order to speed up the convergence, a non-linear element Gauss-Seidel approach is also possible, based on the *quasi* element-Jacobi solver. Hence, a third variant of this solver is the *quasi* non-linear element Gauss-Seidel (qNGS).

$p$	Size of $[D]$
0	$5 \times 5$
1	$20 \times 20$
2	$50 \times 50$
3	$100 \times 100$
4	$175 \times 175$
5	$280 \times 280$
6	$420 \times 420$

**Table 1:** The size of the diagonal matrix  $[D]$  as a function of expansion order ( $p$ ) for tetrahedral element.

## 2.4 The $hp$ -Multigrid Approach

Multigrid methods are known as efficient techniques for accelerating convergence to steady state for both linear and non-linear problems [56, 57], and can be applied with a suitable existing relaxation technique. The rapid convergence property relies on an efficient reduction of the solution error on a nested sequence of coarse grids.

The spectral multigrid approach is based on the same concepts as a traditional  $h$ -multigrid method, but makes use of “coarser” levels which are constructed by reducing the order of accuracy of the discretization,



rather than using physically coarser grids with fewer elements. Thus, all grid levels contain the same number of elements, which alleviates the need to perform complex interpolation between grid levels and/or to implement agglomeration-type procedures [57]. Furthermore, the formulation of the interpolation operators, between fine and coarse grid levels, is greatly simplified when a hierarchical basis set is employed for the solution approximation. The main advantage is due to the fact that the lower order basis functions are a subset of the higher order basis (*i.e.* hierarchical) and the *restriction* and *prolongation* operators become simple projection operators into a lower and higher order space, respectively [58]. Therefore their formulation is obtained by a simple deletion or augmentation of the basis set. The *restriction* from fine to coarse level is obtained by disregarding the higher order modal coefficients and transferring the values of the low order modal coefficients exactly. Similarly, the *prolongation* from coarse to fine levels is obtained by setting the high order modes to zero and injecting the values of the low order coefficients exactly.

Multigrid strategies are based on a recursive application of a two-level solution mechanism, where the second (coarser) grid is solved exactly, and used to accelerate the solution on the finer grid [56]. Because the exact solution of the coarse grid problem at each multigrid cycle is most often prohibitively expensive, the recursive application of multigrid to solve the coarse grid problem offers the preferred approach for minimizing the computational cost of the multigrid cycle, thus resulting in a complete sequence of coarser grids. For spectral ( $p$ )-multigrid methods, the recursive application of lower order discretizations ends with the  $p = 0$  discretization on the same grid as the fine level problem. For relatively fine meshes, the (exact) solution of this  $p = 0$  problem at each multigrid cycle can become expensive, and may impede the  $h$ -independence property of the multigrid strategy. The  $p = 0$  problem can either be solved approximately by employing the same number of smoothing cycles on this level as on the finer  $p$  levels, or the  $p = 0$  problem can be solved more accurately by performing a larger number of smoothing cycles at each visit to this coarsest level. In either case, the convergence efficiency will be compromised, either due to inadequate coarse level convergence, or to excessive coarse level solution cost. An alternative is to employ an  $h$ -multigrid procedure to solve the coarse level problem at each multigrid cycle. In this scenario, the  $p$ -multigrid scheme reverts to an agglomeration multigrid scheme once the  $p = 0$  level has been reached, making use of a complete sequence of physically coarser agglomerated grids, thus the designation  $hp$ -multigrid. Agglomeration multigrid methods make use of an automatically generated sequence of coarser level meshes, formed by merging together neighboring fine grid elements, using a graph algorithm. First-order accurate ( $p = 0$ ) agglomeration multigrid methods for unstructured meshes are well established and deliver near optimal grid independent convergence rates [59]. This procedure has the potential of resulting in a truly  $h$ - and  $p$ -independent solution strategy for high-order accurate discontinuous Galerkin discretizations. Figure (2) illustrates a three dimensional view of a typical two level  $h$ -multigrid agglomerated (AMG) configuration.

## 2.5 Parallel Implementation

In the current-day scientific computing environment, realistic three-dimensional simulations will almost always require the use of massively parallel distributed memory computer architectures. The discontinuous Galerkin  $hp$ -multigrid solver described in this work achieves parallelism through domain decomposition, and uses the standard MPI message-passing library for inter-processor communication [60]. The mesh is partitioned using the METIS graph partitioner [61] operating on the dual graph of the mesh, where the cell centroids represent the graph vertices, and faces delimiting neighboring cells represent the graph edges. The partitioning operation assigns groups of cells to individual partitions, and faces which border on two cells in different partitions are said to be intersected, and are assigned uniquely to one of these partitions,



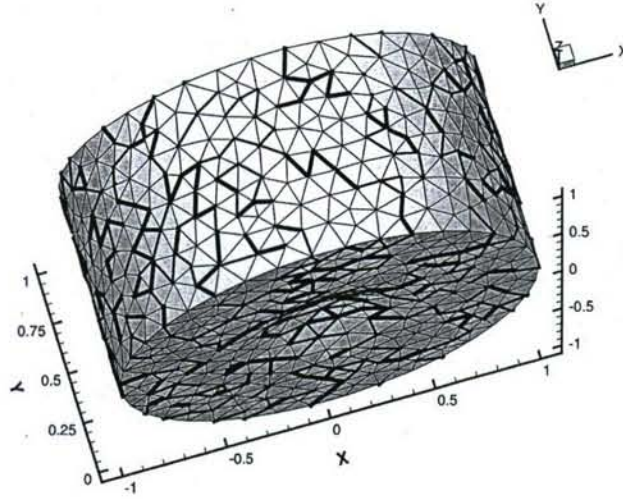


Figure 2: A typical two level  $h$ -multigrid mesh configuration.

while a ghost cell is created in the partition containing the intersected face. This ghost cell corresponds to the neighboring cell in the remote partition for the intersected face, as depicted in Figure 3, and is used as a buffer to temporarily store the flux values computed on the face, which are then sent and accumulated to the real image of the ghost cell in the remote partition using the MPI communication library. This approach entails no duplication of computations in adjacent partitions, and results in the same number of operations and exact same residual values at each iteration as the corresponding sequential algorithm. Standard techniques for masking communication latency are employed, including packing all messages destined for the same remote partition into a single large message, and using non-blocking send and receive calls through the MPI library.

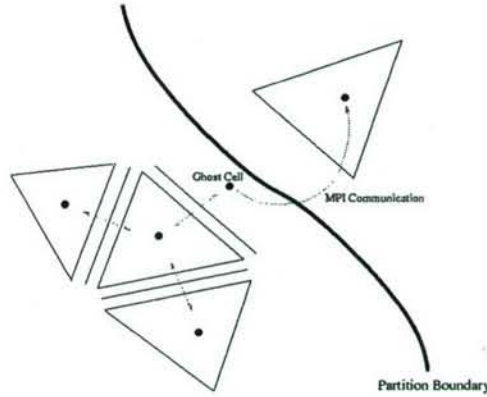
Since the mesh topology is unchanged for the various  $p$  levels of the  $hp$ -multigrid algorithm, the same communication patterns are used for all  $p$  levels. Furthermore, no communication is required in the restriction and prolongation operations between the various  $p$ -multigrid levels.

On the other hand, the agglomerated coarse levels used in the  $h$ -portion of the  $hp$ -multigrid algorithm are all partitioned independently with METIS, and then sorted in order to maximize the overlap between coarse and fine  $h$ -level partitions. Thus, on each agglomerated coarse  $h$ -level, a different communication pattern is used, and additional communication patterns are required for performing the restriction and prolongation between the coarse and fine levels of the  $h$ -multigrid sequence. The partitioning operations and determination of the communication patterns and buffers are performed sequentially and are precomputed and stored prior to the initiation of the flow solution phase.

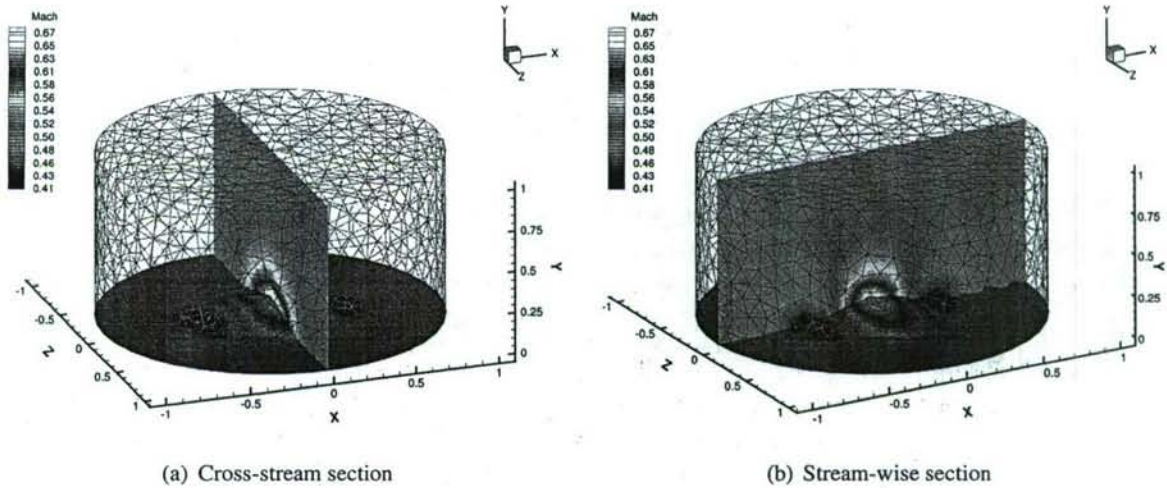
## 2.6 Simple flow configuration

The accuracy of the spatial discretizations and the efficiency of the solution schemes described above are evaluated for the Euler equations using a test problem consisting of the compressible flow over a three-dimensional bump. A series of four grids on this configuration have been generated, consisting of  $N = 1220$ ,





**Figure 3:** Illustration of intersection of mesh elements at inter-processor partition boundary, creation of ghost cell, and communication between ghost cell and real image in remote partition.



**Figure 4:** The Mach number contours for the three-dimensional bump case on 10,349 element mesh using fifth-order accurate ( $p=4$ ) discretization.

5041, and 10349 tetrahedral elements, respectively, in order to study the grid convergence of the discontinuous Galerkin discretizations of various orders. For each case the solution was converged to machine zero in the discretization error studies. Unless otherwise stated, all the simulated results are initiated with a  $p = 0$  solution obtained a priori. The full domain extends from  $-1 \leq x \leq 1$  in the stream-wise direction, from  $0 \leq y \leq 1$  in the vertical direction and from  $-1 \leq z \leq 1$  in the cross-stream direction, with wall boundaries at  $y = 0$ . All the simulation results for this configuration are performed with the HLLC flux using a single processor machine.

### 2.6.1 Accuracy Validation

The aforementioned flow configuration is used to assess the accuracy of the DG discretizations. Figure (4) shows the Mach contour lines for a free-stream Mach number of  $M_\infty = 0.5$ , for the  $p = 4$  (fifth-order) accurate discretization on the finest mesh of 10,349 cells.

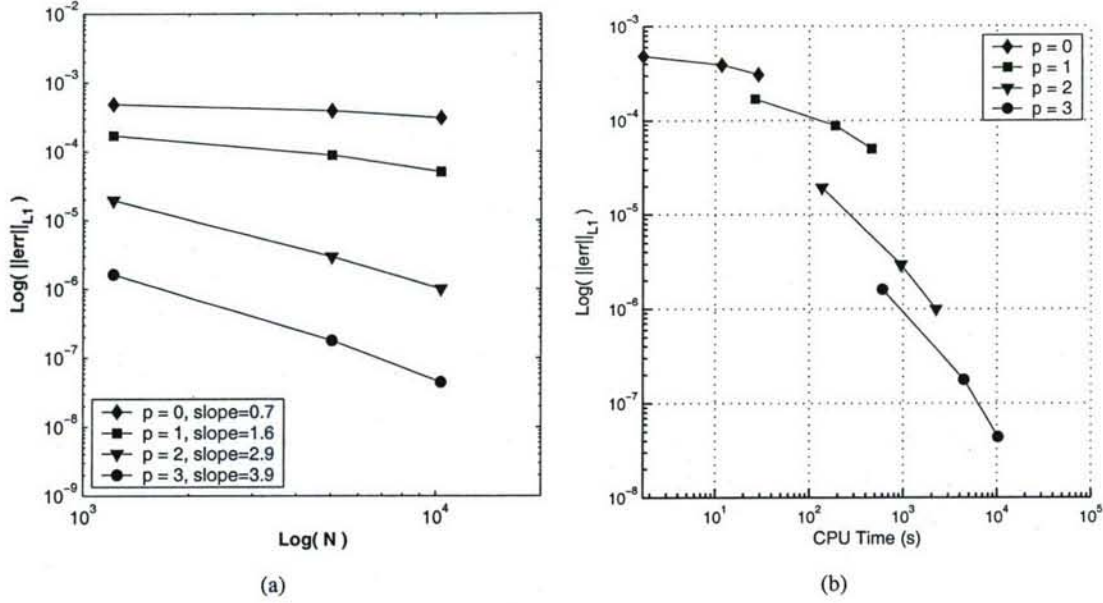


Figure 5: The  $L_1$  norm of the entropy error as a function of: (a)  $h/p$ -refinement; (b) CPU time.

Figure 5(a) shows the accuracy (i.e. the  $L_1$  entropy error norm) of the steady-state solution for 1st, 2nd, 3rd and 4th order accurate discretizations as a function of the number of elements. For three-dimensional configurations the number of elements,  $N$ , is proportional to  $1/h^3$ , where  $h$  represents an approximation of the cell size. The asymptotic slope of these curves indicates that the optimal error convergence rate ( $\approx h^{p+1}$ ) is obtained. A comparison of the computed accuracy versus CPU time is given in Figure 5(b), where the various  $p$ -discretizations have been converged to machine zero on the various grid configurations using the *quasi* linearized element Jacobi driven multigrid scheme. In general, for a given level of accuracy, the CPU time decreases when the approximation order is increased, with the benefit increasing for smaller accuracy tolerances.

### 2.6.2 Multigrid Efficiency Study

In the context of the  $hp$ -multigrid methodology, the same flow configuration is considered with geometrical parameters, boundary and initial conditions as defined in Section (2.6). The multigrid convergence efficiency is studied for this test case using various grid sizes and for different orders of accuracy  $p$ . A multigrid V-cycle is used, which runs through the various  $p$  level discretizations, beginning with the highest  $p$  level, and proceeding to  $p=0$ , followed by visits to the various coarse agglomerated levels, before reversing the process to return to the original highest  $p$  level. The number of  $p$  and  $h$  levels depends on the discretization accuracy and mesh resolution of the problem to be solved. A total of  $p+1$  spectral multigrid levels are used for a  $p$ -discretization problem, and 3, 4 or 5  $h$ -levels (AMG) are considered for mesh sizes of  $N = 1220$ ,  $N = 5041$ ,  $N = 10349$  elements, respectively, using 10 element Jacobi iterations per level, in all cases.

Figure (6)a illustrates the convergence rate for a fixed mesh size of  $N = 10349$  elements and various orders ( $p = 1, 2, 3, 4$ ), while Figure (6)b illustrates the convergence rate for a fixed order of  $p = 4$  over the various coarse and fine meshes. These figures illustrate the  $p$ - and  $h$ - independence of the  $hp$ -multigrid algorithm, with only slight increase in the total number of iterations from 23 to 25 occurring when going



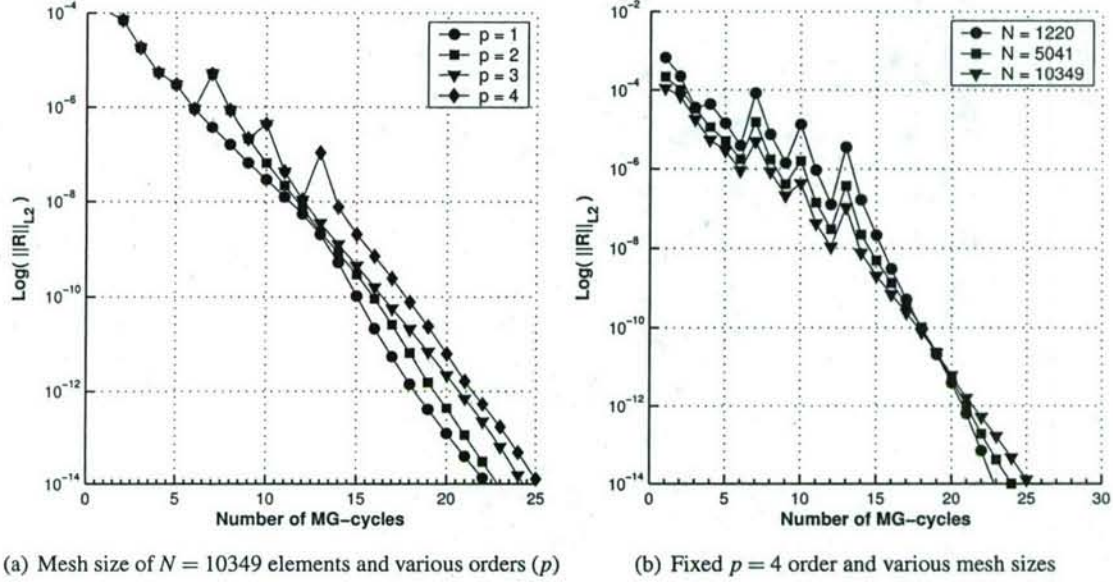


Figure 6: The  $L_2$  norm of the residual vs. the number of multigrid cycles, for the simple flow configuration case.

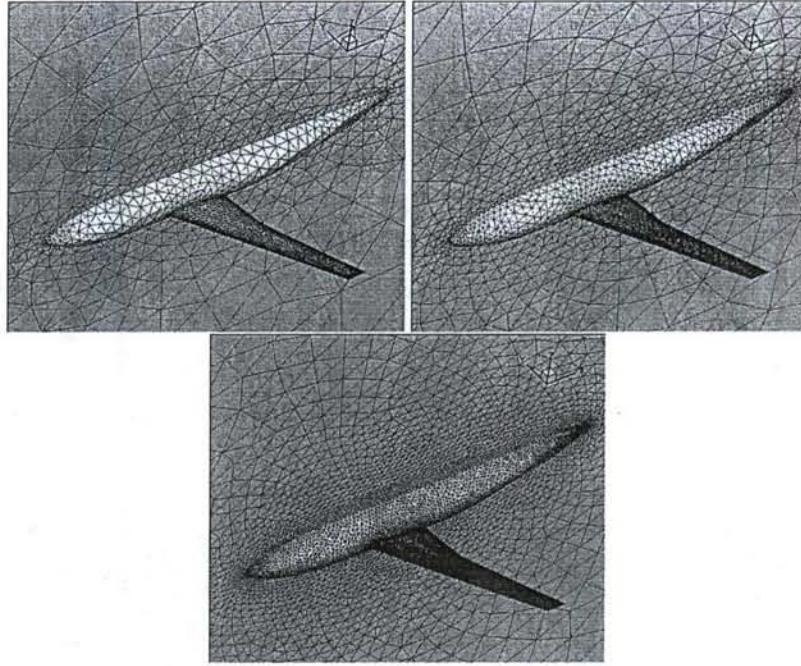
from the 1220 to the 10,349 element mesh. In spite of the good performance of the  $hp$ -multigrid on this test case, the small mesh sizes and simple geometry make for a relatively simple problem, and the evaluation of the solver on more complex configurations is required.

## 2.7 Complex flow configuration

In order to fully assess the performance of the  $hp$ -multigrid DG solver both in terms of speed of convergence and parallel scalability, a more complex test problem consisting of low-speed flow over a three-dimensional DLR-F6 wing-body configuration is considered. A series of three coarse to fine grids was generated on this configuration using the VGRID grid generation program [51]. The coarse, medium and fine meshes contain approximately 185,000, 450,000 and 2.6 million tetrahedral cells, respectively and are depicted in Figure 7. The agglomeration procedure was used to construct 4 coarse levels for the  $N = 185,000$  mesh, 5 coarse levels for the  $N = 450,000$  mesh, and 6 coarse levels for the  $N = 2,600,000$  mesh, for use in the h-multigrid portion of the solver. These meshes were then partitioned and the computations were performed on an in-house Linux cluster, as well as on the NASA Columbia Supercomputer.

A full multigrid variant of the  $hp$ -multigrid solver was used exclusively. This is initiated by initializing a  $p=0$  solution using freestream values, and approximately solving this low-order problem using a small number of h-multigrid cycles. The resulting flowfield is then used as the initial condition for the solution of a  $p=1$  problem, which is solved using the  $hp$ -multigrid scheme, and the process is repeated recursively until the desired  $p$ -level solution is reached. Ten multigrid cycles are used at each  $p$ -level in the full multigrid procedure, prior to reaching the last level, after which of the order of 100 multigrid cycles are used to fully converge the final highest  $p$ -level problem. Each multigrid cycle, in turn, consists of a V-cycle with 10 quasi-non-linear element Jacobi cycles performed on the coarsening phase of the cycle. The freestream Mach number is taken as  $\text{Mach}=0.5$ , and the incidence is zero degrees, in order to avoid the formation of any shock waves for this case.

For large three-dimensional problems of this type, the use of the agglomerated h-grid levels is crucial for maintaining rapid convergence rates, since the low-order  $p=0$  problem itself will be slowly converging in the



**Figure 7:** Coarse (185,000 cells), medium (450,000 cells) and fine (2.6M cells) mesh for wing body configuration.

absence of multigrid. This is demonstrated in Figure 8, where the convergence history of the  $p=0$  problem on the coarse, medium and fine grid is depicted with and without the multigrid algorithm. In the single level (non-multigrid) case, the solution of the  $p=0$  problem is seen to require a large number of iterations, which grows substantially as the mesh resolution is increased, reaching over 3000 cycles for the finest 2.6M cell mesh. On the other hand, the  $h$ -multigrid scheme produces consistent grid independent convergence rates for this problem in approximately 50 multigrid cycles. The convergence properties of the  $p=0$  problem in terms of number of cycles and cpu time are summarized in Table (2).

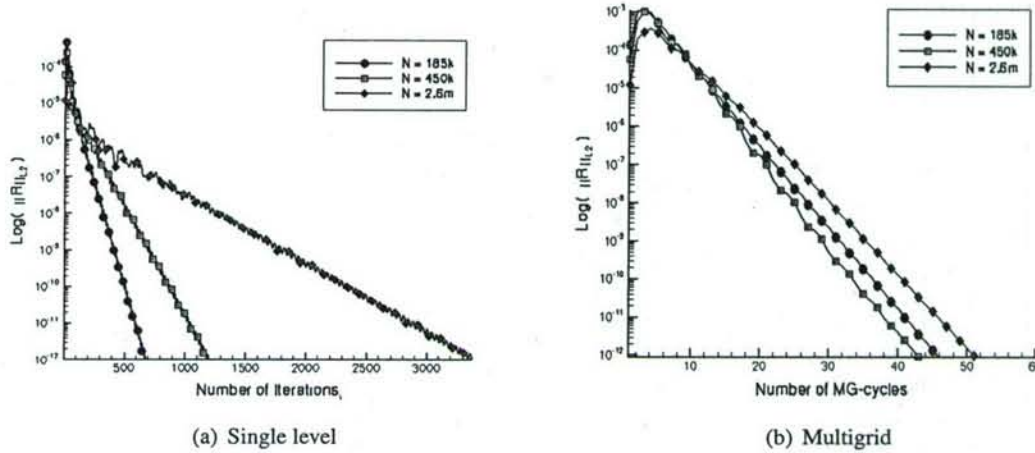
$N$	Single grid	AMG	No. of AMG Levels
185k	679	46	4
450k	1200	43	5
2.6m	3375	51	6

**Table 2:** Convergence in terms of number of iterations without and with multigrid for fixed order  $p = 0$  on coarse, medium and fine meshes.

### 2.7.1 Multigrid Results

Figures 9 and 10 illustrate the convergence histories achieved on the three DLR-F6 wing-body meshes for various  $p$  orders of accuracy using the full  $hp$ -multigrid scheme. The peaks in the convergence plots correspond to the transition from low to higher-order discretizations in the initial phase of the full multigrid scheme. At each transition, the residual increases suddenly as a new higher-order discretization is initialized, but quickly returns to the level observed by continuing the multigrid iterations on the original  $p$ -order discretization. Thereafter, the higher-order discretization resumes convergence at a similar rate to the lower





**Figure 8:** Comparison of convergence of  $p = 0$  discretization (a) without and (b) with  $h$ -multigrid for coarse, medium and fine meshes for wing-body configuration.

order discretizations.

The plots in Figure 9 reveal essentially  $p$ -independent convergence rates for all three grids, except for the  $p=4$  discretization which is slower to converge on the coarse 185,000 cell grid. On the finer grids, this discretization exhibited poor robustness and could not be made to converge adequately, while the  $p=3$  discretization failed on the finest grid and was only run successfully on the coarse and medium grids.

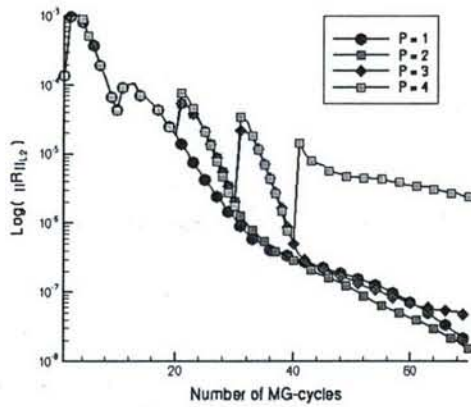
Figure 10 compares the convergence rates for  $p=1$  and  $p=2$  on all three meshes (since the higher  $p$  discretizations failed to run on the finer meshes), showing nearly  $h$ -independent convergence rates going from 180,000 to 2.6 million cell meshes. In all cases, the residuals are reduced 5 to 7 orders of magnitude in under 100  $hp$ -multigrid cycles, by which time the lift coefficient has attained its final converged value.

The solution in terms of computed Mach contours is displayed in Figure 11 for various  $p$  orders of accuracy and on the coarse and fine mesh, illustrating the high accuracy achievable on relatively coarse meshes using high-order DG discretizations.

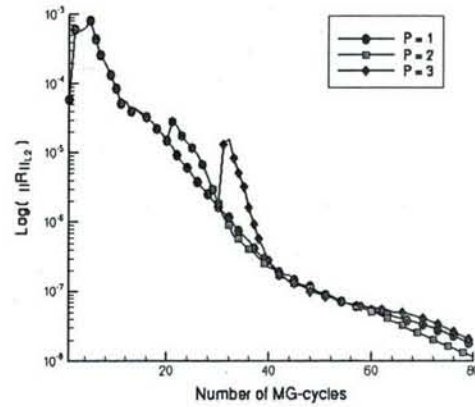
## 2.7.2 Computational Performance

The parallel performance of the  $hp$ -multigrid algorithm has been assessed by examining the scalability of this solver for various orders of accuracy (from  $p=0$  to  $p=4$ ) on the coarse (185,000 cells) and fine (2.6M cells) meshes, running on the NASA Columbia machine, using from 32 to 2008 cpus. The NASA Columbia Supercluster consists of a collection of loosely coupled SGI Altix nodes, each containing 512 tightly connected cpus. Four of these nodes are tightly coupled through a shared NUMALink4 interconnect, providing a global shared memory image and high bandwidth over 2048 cpus [62]. While the global shared memory image is not required in our case, due to the MPI implementation of the parallel  $hp$ -multigrid solver, the high bandwidth interconnect is helpful for achieving good scalability on more than 512 processors.

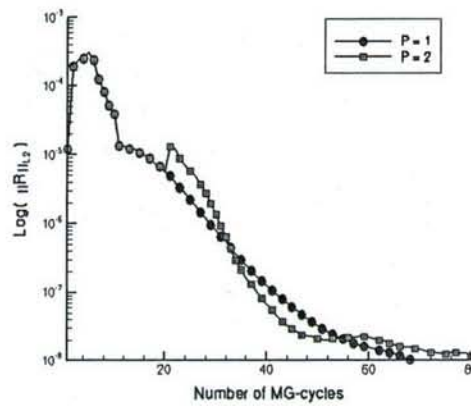
Figure 12a depicts the parallel speedup achieved for the  $hp$ -multigrid algorithm running on the coarse mesh of 185,000 cells, going from 32 to 1004 processors, where the 32 processor runs are assumed to exhibit perfect speedup, in view of the fact that runs on smaller number of processors were not performed for lack of memory at the higher orders of accuracy ( $p$ ). In all cases, three  $h$ -multigrid levels are employed,



(a) Coarse mesh: 185,000 cells



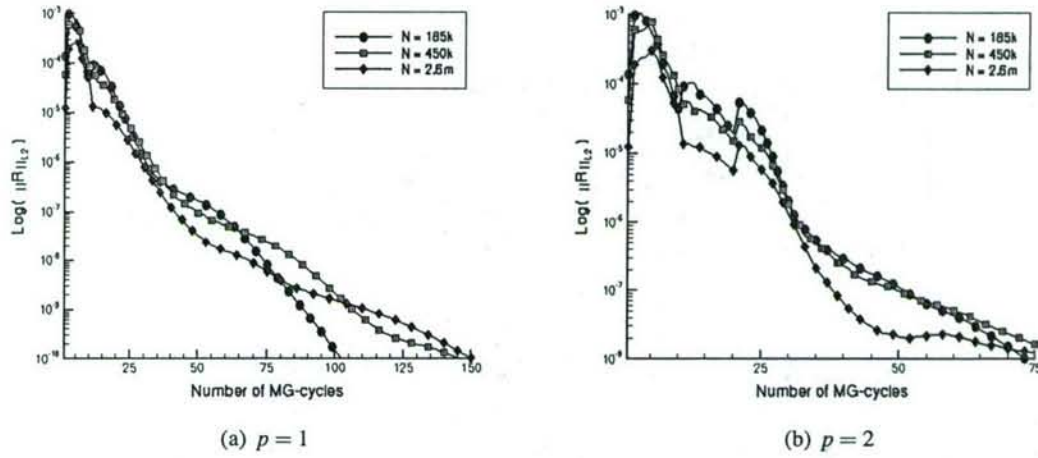
(b) Medium mesh: 450,000 cells



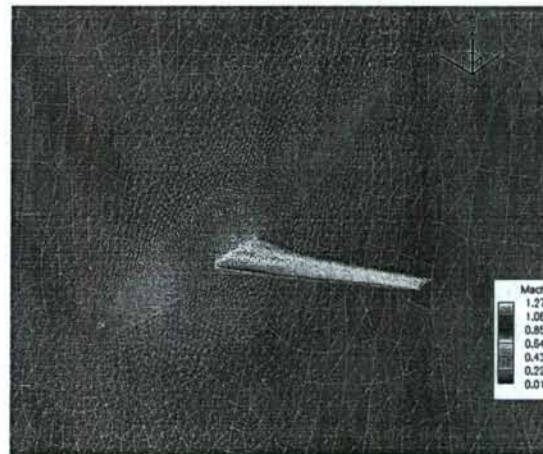
(c) Fine mesh: 2.6 million cells

**Figure 9:** The  $L_2$  norm of the residual vs. number of  $hp$ -multigrid cycles for compressible flow solution on wing-body configuration using various discretization orders and on various size meshes.





**Figure 10:** Variation of convergence histories for  $p = 1$  and  $p = 2$  discretizations with grid resolution as measured by the  $L_2$  norm of the residual vs. number of  $hp$ -multigrid cycles on the various meshes.



(a)  $p = 1$

**Figure 11:** Computed solution in terms of Mach contours on DLR-F6 wing body configuration.



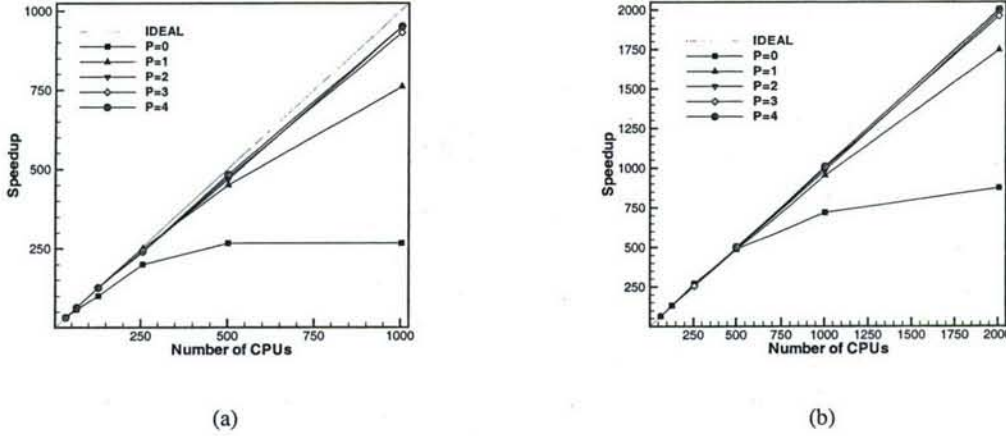


Figure 12: The speedup vs. the number of processors for: (a) coarse mesh; (b) fine mesh

with the appropriate number of  $p$  levels depending on the overall order of accuracy of the finest level. For  $p=0$  (i.e. first-order accurate), which corresponds to the traditional agglomeration  $h$ -multigrid applied to a cell-centered finite volume scheme, scalability is seen to fall off dramatically as the number of processors is increased. This is to be expected, since the number of elements is relatively small in this case. For example, on 1004 cpus, the number of mesh cells per processor is less than 185 for this case. Thus, the ratio of computation to communication is extremely small in such cases. However, as the order of accuracy is increased, the scalability improves dramatically. For example, the second-order accurate  $p=1$  discretization achieves a speedup of 750 on 1004 cpus, while the higher order discretizations achieve speedups close to 950 on 1004 cpus. Results are depicted in Figure 12b for the fine 2.6 million cell mesh using up to 2008 cpus. Similar trends are observed, with the  $p=0$  results showing a significant drop-off in scalability, while the highest order results show near perfect scalability up to 2008 cpus, due to the increased computation to communication ratio for this finer mesh.

On 2008 cpus, a single multigrid iteration at  $p=4$  required 83 seconds of wallclock time, while the same multigrid iteration at  $p=1$  required only 1.2 seconds. This translates into an overall solution time of under 2 minutes for the  $p=1$  case on the fine mesh, using approximately of the order of 100 multigrid cycles and potentially 2 hours for the  $p=4$  case, although the  $p=4$  (and  $p=3$ ) cases were not run to completion on the fine mesh due to robustness issues. Note that the  $p=4$  case on the finest grid corresponds to a problem with over 455 million degrees of freedom.

## 2.8 Extension to Implicit Time-Dependent Problems

While the above results demonstrate the efficiency of the multigrid method for steady-state problems, many problems of interest are unsteady in nature and require the resulting spatially discretized equations to be integrated in time. Although the use of explicit time-integration schemes has been widespread for DG discretizations, in this work we focus on the use of implicit time-integration schemes, which are not restricted by the CFL-stability limit of explicit methods, and thus are capable of using maximum time-steps determined by accuracy considerations, and are thus more suitable for stiff problems. The implicit time-integration schemes currently employed in this work range from first to fourth-order accurate in time, including both first and second-order accurate multistep backwards difference formulations (BDF1, BDF2),

the second-order accurate Crank-Nicolson or trapezoidal scheme, and a fourth-order accurate implicit multistage Runge-Kutta scheme (IRK4). The strategy used in this project has been to extend the h-p multigrid method for use in solving the implicit system arising at each time step from these time discretizations. For the scope of this project, the time-dependent investigations have been limited to two dimensional inviscid problems.

Because we are concerned with obtaining highly accurate solutions, the high spatial accuracy must be matched with high temporal accuracy, which can be achieved either by using small time steps, or by resorting to higher-order accurate time discretizations. However, stability of time-discretization schemes is closely related to their formal order of accuracy. The BDF1 and BDF2 schemes are both unconditionally stable, while the CN2 scheme is A-stable, but not L-stable [63]. For these reasons, CN2 has often been shunned in favor of BDF2 in many computational fluid dynamics problems. However, the lack of L-stability may be acceptable particularly for problems with smooth solutions, and the scheme is therefore included in the present study. Because higher-order multistep backwards difference schemes beyond second-order are not A-stable, we choose to investigate the use of implicit Runge Kutta schemes for achieving higher temporal accuracy. A six-stage diagonally implicit IRK scheme is chosen which is fourth-order accurate in time. While this scheme may not necessarily represent the optimal fourth-order temporal scheme for all problems, it has been designed with stiff stability and accuracy considerations in mind [64], and has been used successfully on lower-order finite volume schemes by various authors [65, 66]. One of the drawbacks of IRK methods is their expense, since these require the solution of multiple implicit problems at each time step (one per stage), as opposed to BDF and CN schemes which only require the solution of a single implicit problem per time step. Therefore, one of the objectives of this work is to determine if IRK schemes can be competitive or superior to lower-order schemes when used in conjunction with efficient solvers, particularly when high accuracy is required.

Starting from the set of ordinary differential equations given by equation (13), the formulations for BDF1, BDF2 and CN2 schemes are given respectively as:

$$\frac{M}{\Delta t} (\mathbf{u}_h^{n+1} - \mathbf{u}_h^n) + \mathbf{R}_p(\mathbf{u}_h^{n+1}) = 0 \quad (18)$$

$$\frac{M}{\Delta t} \left( \frac{3}{2} \mathbf{u}_h^{n+1} - 2\mathbf{u}_h^n + \frac{1}{2} \mathbf{u}_h^{n-1} \right) + \mathbf{R}_p(\mathbf{u}_h^{n+1}) = 0 \quad (19)$$

$$\frac{M}{\Delta t} (\mathbf{u}_h^{n+1} - \mathbf{u}_h^n) + \frac{1}{2} \mathbf{R}_p(\mathbf{u}_h^{n+1}) + \frac{1}{2} \mathbf{R}_p(\mathbf{u}_h^n) = 0 \quad (20)$$

where  $\Delta t$  represents the integration time step, and  $\mathbf{u}_h^n$  and  $\mathbf{u}_h^{n+1}$  are numerical solutions for the current and the next (unknown) time step, respectively. By defining a nonlinear unsteady residual,  $\mathbf{R}_e$ , for the corresponding BDF1, BDF2 and CN2 schemes as:

$$\text{BDF1 : } \mathbf{R}_e(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t} \mathbf{u}_h^{n+1} + \mathbf{R}_p(\mathbf{u}_h^{n+1}) - \frac{M}{\Delta t} \mathbf{u}_h^n = 0 \quad (21)$$

$$\text{BDF2 : } \mathbf{R}_e(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t} \left( \frac{3}{2} \mathbf{u}_h^{n+1} \right) + \mathbf{R}_p(\mathbf{u}_h^{n+1}) - \frac{M}{\Delta t} (2\mathbf{u}_h^n - \frac{1}{2} \mathbf{u}_h^{n-1}) = 0 \quad (22)$$

$$\text{CN2 : } \mathbf{R}_e(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t} \mathbf{u}_h^{n+1} + \frac{1}{2} \mathbf{R}_p(\mathbf{u}_h^{n+1}) - \left( \frac{M}{\Delta t} \mathbf{u}_h^n - \frac{1}{2} \mathbf{R}_p(\mathbf{u}_h^n) \right) = 0 \quad (23)$$

the solution of these schemes can be achieved by solving the nonlinear problems  $\mathbf{R}_e(\mathbf{u}_h^{n+1}) = 0$  at each time step.



These schemes are relatively efficient because they solve only one implicit set of equations per time step. In the case of multistage IRK schemes, however, multiple implicit problems are required per time step (each stage works on one implicit system solution), but these schemes are easily implemented in the presence of variable time steps and can be constructed to be A- and L-stable for any temporal order. In this paper, we utilize the ESDIRK class of RK schemes, which corresponds to Explicit first stage, Single Diagonal coefficient, diagonally Implicit Runge-Kutta. The formula for a  $S$ -stage ESDIRK scheme, in the case of the Euler equations, can be written as,

$$\begin{aligned} (i) \quad & \mathbf{u}^{(0)} = \mathbf{u}_h^n \\ (ii) \quad & \text{For } s = 1, \dots, S \\ & \mathbf{u}^{(s)} = \mathbf{u}^n - \Delta t \sum_{j=1}^s a_{sj} M^{-1} \mathbf{R}_p(\mathbf{u}^{(j)}) \\ (iii) \quad & \mathbf{u}_h^{n+1} = \mathbf{u}^{(S)} \end{aligned} \tag{24}$$

where  $a_{sj}$  are the Butcher coefficients of the scheme. The Butcher table for the six-stage ESDIRK scheme ( $S = 6$ , fourth-order accurate) employed presently is shown in Table 3 and the values are given in references [65, 66]. The set of coefficients,  $a_{sj}$ , defines the implicit RK schemes (as shown in Eq.(24)). The first stage is explicit due to  $a_{11} = 0$ . A single implicit scheme is solved at each individual stage since the set of  $a_{kj}$  has the form of a lower triangular matrix. The solution in the last stage is the solution for the next time step and  $c_k$  represents the point in the time interval,  $[t, t + \Delta t]$  and satisfies,

$$c_k = \sum_{j=1}^k a_{kj} \quad (k = 1, 2, \dots, 6) \tag{25}$$

Similarly, the unsteady residual corresponding to the non-linear implicit system at each stage of the IRK4 scheme can be written as,

$$\begin{aligned} \mathbf{R}_e(\mathbf{u}^{(s)}) &= \frac{M}{\Delta t} \mathbf{u}^{(s)} + a_{ss} \mathbf{R}_p(\mathbf{u}^{(s)}) \\ &- \left[ \frac{M}{\Delta t} \mathbf{u}^n - \sum_{j=1}^{s-1} a_{sj} \mathbf{R}_p(\mathbf{u}^{(j)}) \right] = 0 \quad (s = 1, \dots, 6) \end{aligned} \tag{26}$$

The basic approach is to modify the h-p multigrid algorithm for use in driving the unsteady residuals defined in equations (21), (22), (23), and (26) to zero, in the place of the steady state residual, at each physical time step.

Two test cases are used to illustrate the performance of the proposed solution techniques for the time-dependent compressible Euler equations. The first test case consists of an isentropic convecting vortex on a uniform mesh, while the second test case consists of the time-dependent vortex shedding from a triangular wedge on a highly graded unstructured mesh.

### 2.8.1 Vortex Convection Case

The convection of a 2-D inviscid isentropic vortex [67–69] is simulated to examine the performance of the proposed implicit time-stepping schemes. The exact solution for this test case at any time  $t$  is the

**Table 3:** Butcher Tableau for ESDIRK class of six-stage RK schemes

$c_1 = 0$	0	0	0	0	0	0
$c_2$	$a_{21}$	$a_{22} = a_{66}$	0	0	0	0
$c_3$	$a_{31}$	$a_{32}$	$a_{33} = a_{66}$	0	0	0
$c_4$	$a_{41}$	$a_{42}$	$a_{43}$	$a_{44} = a_{66}$	0	0
$c_5$	$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$	$a_{55} = a_{66}$	0
$c_6 = 1$	$a_{61} = b_1$	$a_{62} = b_2$	$a_{63} = b_3$	$a_{64} = b_4$	$a_{65} = b_5$	$a_{66}$
$\mathbf{u}^{n+1}$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$

initial solution at  $t_0 = 0$  translated over a distance  $u_\infty t$  for a horizontally convecting vortex, which provides a valuable reference for measuring the accuracy of the computed solution. The mean flow density,  $\rho_\infty$ , velocity,  $u_\infty$  and  $v_\infty$ , pressure,  $p_\infty$  and temperature  $T_\infty$  are taken as freestream values, which are set as  $(\rho_\infty, u_\infty, v_\infty, p_\infty, T_\infty) = (1, 0.5, 0, 1, 1)$  in this test case. Freestream boundary conditions are imposed on the top and bottom boundaries, while periodic boundary conditions are applied between the inlet and outlet of the domain. These boundary conditions are applied on all levels of the multigrid sequence. At  $t_0 = 0$ , the flow is perturbed by an isentropic vortex  $(\delta u, \delta v, \delta T)$  centered at  $(x_0, y_0)$  with the form:

$$\delta u = -\frac{\alpha}{2\pi}(y - y_0)e^{\phi(1-r^2)} \quad (27)$$

$$\delta v = \frac{\alpha}{2\pi}(x - x_0)e^{\phi(1-r^2)} \quad (28)$$

$$\delta T = -\frac{\alpha^2(\gamma - 1)}{16\phi\gamma\pi^2}e^{2\phi(1-r^2)} \quad (29)$$

where,  $\phi$  and  $\alpha$  are parameters which determine the strength of the vortex,  $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$  is the distance to the vortex center, and  $\gamma = 1.4$  is the ratio of specific heats of air. In this study, we set  $\phi$  as unity and  $\alpha$  as 4.0. Given the perturbation functions shown in Eq. (27), (28) and (29), we can determine the other resulting conservative variables, assuming isentropic flow throughout the domain (i.e.  $p/\rho^\gamma = 1$  and  $T = p/\rho$  for a perfect gas):

$$\rho = T^{1/(\gamma-1)} = (T_\infty + \delta T)^{1/(\gamma-1)} = \left[1 - \frac{\alpha^2(\gamma-1)}{16\phi\gamma\pi^2}e^{2\phi(1-r^2)}\right]^{1/(\gamma-1)} \quad (30)$$

$$u = u_\infty + \delta u = 0.5 - \frac{\alpha}{2\pi}(y - y_0)e^{\phi(1-r^2)} \quad (31)$$

$$v = v_\infty + \delta v = 0 + \frac{\alpha}{2\pi}(x - x_0)e^{\phi(1-r^2)} \quad (32)$$

This test case employs a uniform Cartesian triangular grid. The initial vortex is placed at  $(x_0, y_0) = (0, 0)$  on a domain of  $-7 \leq x \leq 7$  and  $-3.5 \leq y \leq 3.5$  with 10000 elements, as shown in Fig. 13. A fifth-order accurate ( $p = 4$ ) spatial discretization is used in all cases, and the time-step ( $\Delta t$ ) is set equal to 0.2. Since the local CFL number is defined as,



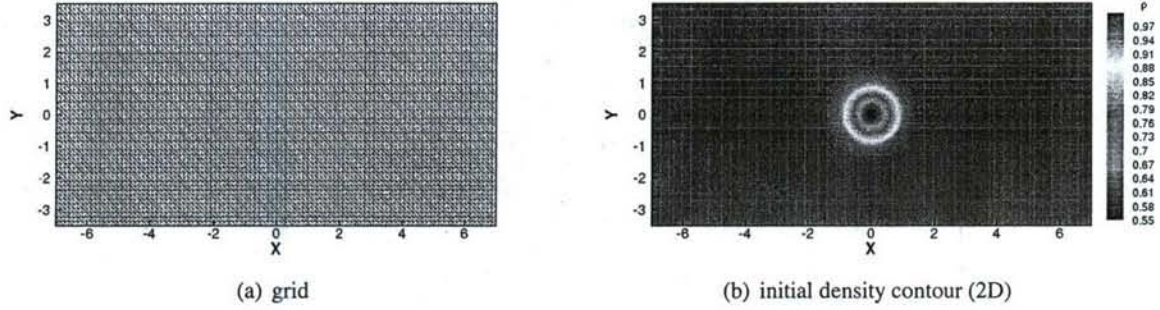


Figure 13: Grid and initial density contours for isentropic vortex convection problem

$$CFL_i = \frac{\Delta t}{vol_i} \sum_{j=1}^{3edges} (|\mathbf{u} \cdot \mathbf{n}| + c)_j \quad 1 \leq i \leq nElem \quad (33)$$

where  $vol$  denotes the area of the element in the 2D case and  $c$  is the local speed of sound, then the fixed time-step  $\Delta t = 0.2$  corresponds to a maximum CFL number of 11. Note that for the considered spatial discretization ( $p=4$ , fifth-order accurate), the explicit stability limit could be as much as 176 times smaller (i.e.  $\sim \frac{1}{p^2}$ ) than our chosen time step.

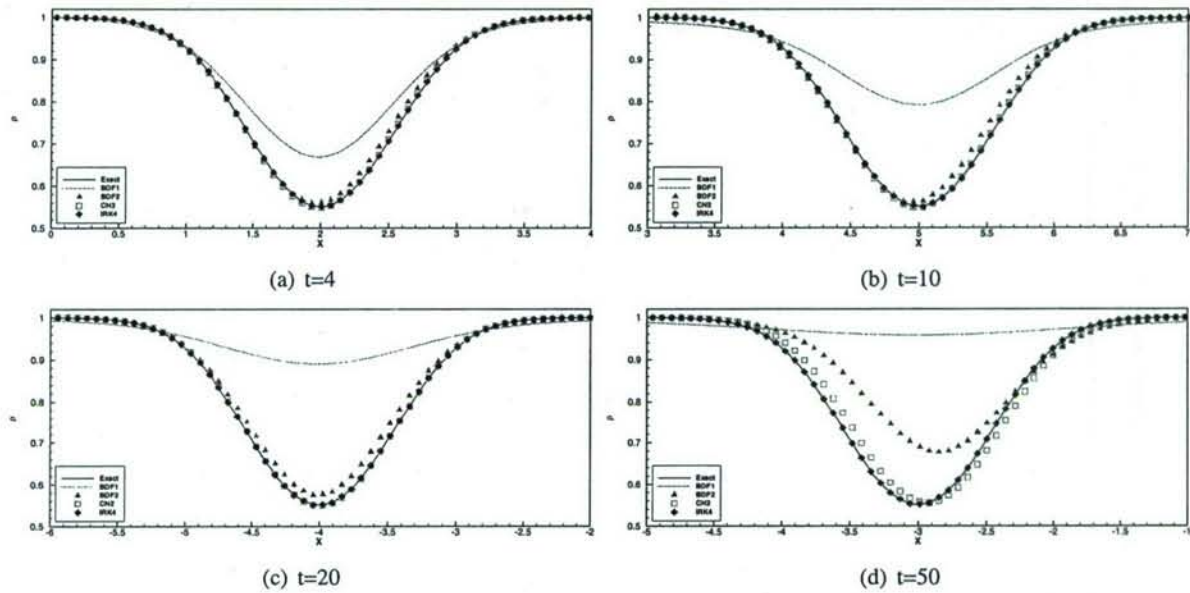
We begin by examining the error of the respective time-integration schemes at a fixed time-step size, in order to determine if these schemes are within or close to their asymptotic regions of convergence. This is important since the error properties of these schemes are asymptotic in nature, based on the presumption of smooth solutions, and there is no guarantee that higher-order methods will deliver smaller errors than lower order methods, even at equivalent time steps, when these assumptions do not hold.

Fig. 13 illustrates the computational mesh and the initial density contours in the domain. The length of the domain is 14, and the horizontal velocity is  $u_\infty = 0.5$ , thus the vortex requires  $T = 28$  to complete one revolution around the periodic grid in the x-direction.

A quantitative comparison is given in Fig. 14, where the density profiles along the horizontal centerline ( $y = 0$ ) for the BDF1, BDF2, CN2 and IRK4 time-integration schemes at various times,  $t = 4$ ,  $t = 10$ ,  $t = 20$ , and  $t = 50$ , are compared with the exact solution, obtained by translating the initial centerline density profile to the appropriate spatial location for each given time. From the figure, it can be seen that the IRK4 scheme exhibits the best resolution and provides very good agreement with the exact solution, since there is no visual deviation between the computed results and exact results and the vortex core is well conserved. On the other hand, the BDF1 scheme which is only first-order accurate in time, produces rapid dissipation of the vortex core, as mentioned previously, while the BDF2 and CN2 schemes provide substantially better resolution and higher accuracy than the BDF1 scheme. Before  $t = 20$ , the computed results obtained by the BDF2 and CN2 schemes fall almost on top of the exact solution, but at later times, these schemes show increased deviations from the exact profile, although the CN2 scheme remains substantially more accurate than the BDF2 scheme. However, this result must be balanced by the fact that the CN2 scheme is not L-stable [63], and may perform poorly in other cases.

In order to assess the asymptotic behavior of the time-integration schemes, a temporal refinement study is carried out using the same fixed spatial discretization ( $p = 4$ ). Because the overall error is due to both spatial and temporal error, the "exact" numerical solution for each temporal scheme is obtained using a small time-step reference solution, in order to eliminate the effect of spatial error and to isolate the temporal





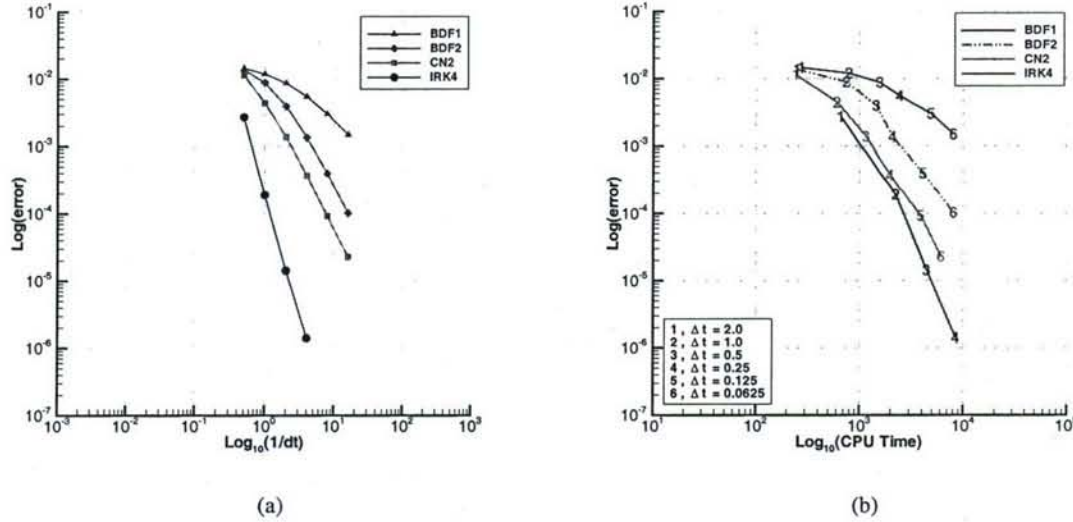
**Figure 14:** Comparison of density profiles for the BDF1, BDF2, CN2 and IRK4 schemes at  $t=4$ ,  $t=10$ ,  $t=20$  and  $t=50$ , using a time-step of  $\Delta t=0.2$

error. The time-step to obtain the "exact" solution is  $\Delta t = 0.01$  for all time-integration schemes. Various time-steps, consisting of  $\Delta t = 2.0$ ,  $1.0$ ,  $0.5$  and  $0.25$ , which correspond to a maximum CFL number of 110, 55, 28 and 14, respectively, have been used for all of the temporal schemes. Additionally, two smaller time-steps,  $\Delta t = 0.125$  and  $0.0625$  are employed for the BDF1, BDF2 and CN2 schemes to extend their range of comparison. A regular mesh with a grid spacing of  $\Delta x = \Delta y = 0.25$  and a total of 3136 elements is employed for this study. The temporal error is obtained by computing the RMS difference of all conserved variables at all grid points between the computed solution and the reference exact solution.

The temporal accuracy results for the BDF1, BDF2, CN2 and IRK4 schemes at  $t = 4$  are illustrated in Fig. 15(a), where the computed temporal error is plotted as a function of the time-step on a log-log plot. The first-order backwards differencing scheme displays a slope of 1.0. The second-order backwards differencing scheme and the Crank-Nicholson scheme demonstrate similar slopes of 1.9 and 2.0, respectively, although the CN2 scheme is consistently more accurate in absolute terms than the BDF2 scheme. The fourth-order Runge-Kutta scheme exhibits a slope of 3.82, which is close to the design value of 4. For any given time-step size, the IRK4 scheme achieves higher accuracy than the BDF1, BDF2 and CN2 schemes, while the BDF2 and CN2 schemes provide better accuracy than BDF1. For example, using a time-step size of  $\Delta t = 0.25$ , the IRK4 scheme attains a temporal error of approximately  $10^{-6}$  while all the other schemes incur temporal errors larger than  $3 \times 10^{-4}$ .

These results demonstrate that the chosen temporal discretization schemes achieve their design order of accuracy within the range of time steps of interest and for high-order spatial discretizations. On the other hand, since the higher-order temporal schemes (particularly the IRK scheme) are more expensive than the lower order schemes, the more practical consideration of computational efficiency of these schemes for a prescribed error tolerance must be addressed. This is done in Figure 15(b), where the accuracy achieved by each implicit time integration scheme is plotted versus cpu time in the place of time-step size. The differences between the various schemes are now reduced compared to the comparisons based on time step





**Figure 15:** (a) Comparisons of temporal accuracy for various implicit temporal schemes as a function of time-step size at  $t = 4$ . (b) Comparisons of temporal accuracy for various implicit temporal schemes as a function of cpu time.

size, due to the fact that the more accurate schemes such as the implicit Runge-Kutta scheme requires five implicit solves per time step, compared to the BDF and CN2 schemes. Nevertheless, as the accuracy requirements are increased, the asymptotic nature of the IRK scheme, where error decreases fastest versus time step size, ensures that this scheme becomes the most efficient overall for small error tolerances.

### 2.8.2 Shedding flow over a triangular wedge

The next test case involves the problem of vortex shedding over a triangular wedge. Because of the inviscid nature of the flow simulation, a triangular geometry is chosen in order to ensure that vortices are produced due to separation at sharp corners. In addition to representing a more realistic test case, involving a highly graded unstructured mesh, this case is also used to study the performance of the various temporal schemes in combination with low and higher-order spatial discretizations, thus focusing on the interplay between spatial and temporal errors. The geometry consists of a triangular wedge placed on the centerline  $y = 0$  of the computational domain, which contains 10,836 unstructured triangular elements, with the ratio of the smallest to largest cell area being 1425:1 (which corresponds to an explicit CFL ratio of 38:1). The flow is inviscid, and a uniform freestream Mach number of 0.2 is applied as the initial condition. Since our principal focus in these calculations is the ability of these schemes in retaining the shape of the vortices as these are convected downstream from the wedge, and in order to reduce the occurrence of initial discontinuities near the surface of the wedge which would be produced with a uniform freestream initial condition and would be detrimental for high-order spatial discretizations, we first employ a  $p = 0$  spatial discretization and BDF2 temporal scheme, utilizing a uniform-flow initial condition:  $u(\mathbf{x}, t = 0) = u_\infty = 0.5$ ,  $v(\mathbf{x}, t = 0) = v_\infty = 0$ ,  $\rho(\mathbf{x}, t = 0) = \rho_\infty = 1.0$  (as shown in Fig. 16(a)), to obtain an intermediate solution (shown in Fig. 16(b)), in which the formed vortices have not yet separated. Then, this intermediate solution is applied as the initial condition to all other high-order  $p = 1$  and  $p = 3$  spatial-discretization schemes, using either the BDF2 or IRK4 time-integration schemes.

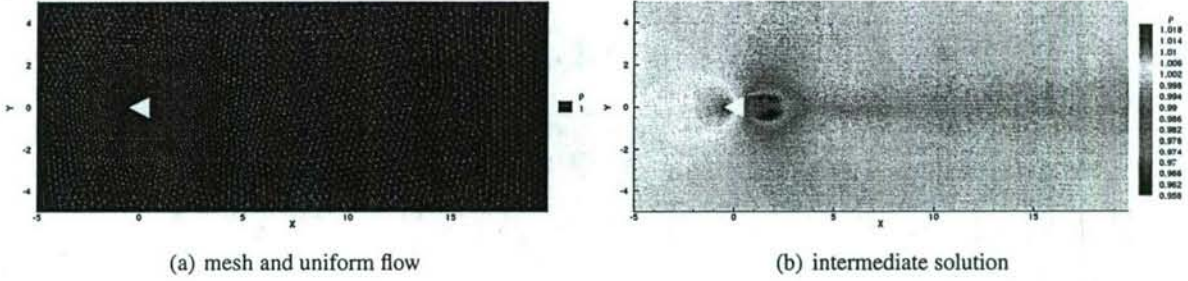


Figure 16: Mesh and density contours for uniform flow and intermediate solution obtained using  $p = 0$  scheme

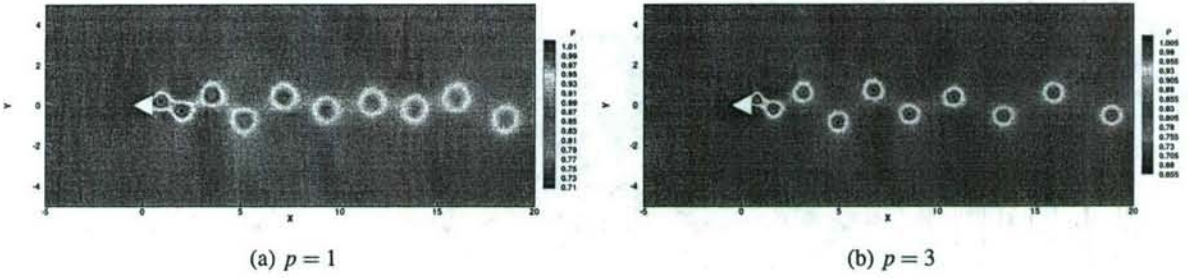


Figure 17: Density contours of  $p=1$  and  $p=3$  spatial-discretization schemes, at  $t=100$ , using BDF2 scheme and  $\Delta t = 0.05$

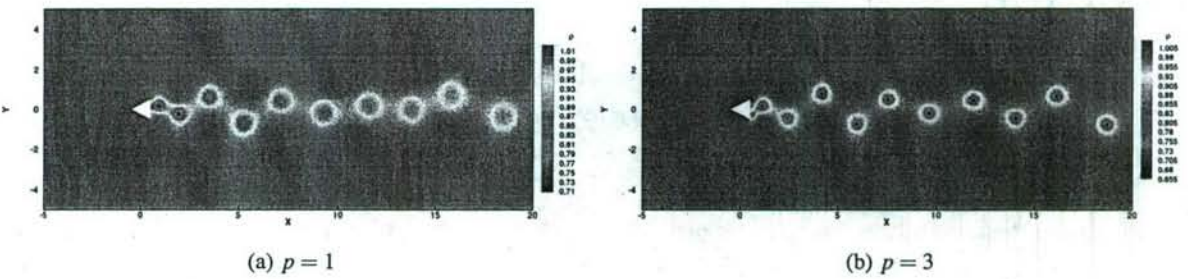
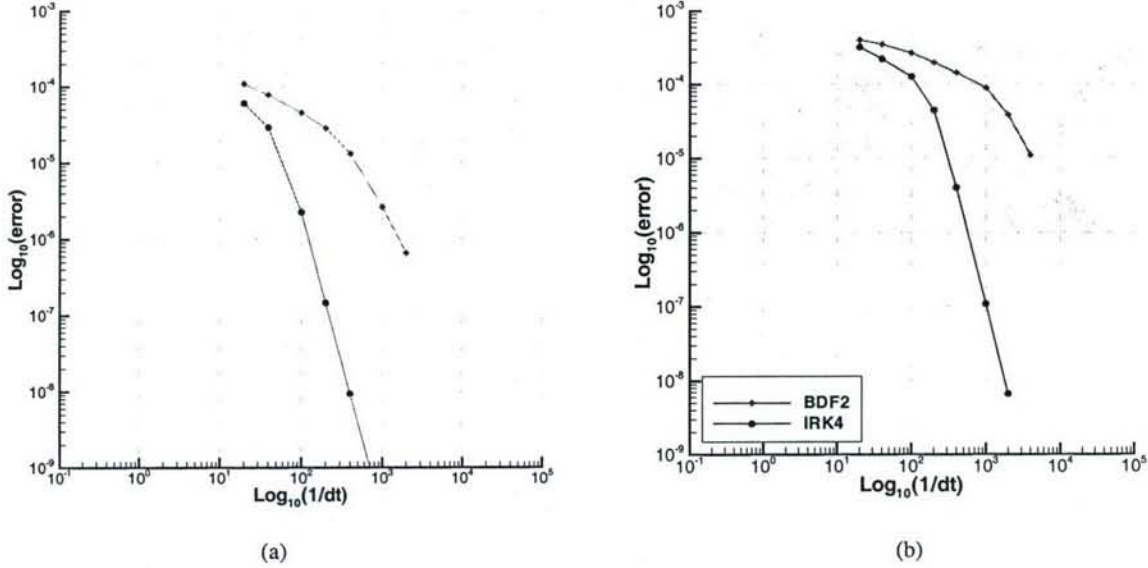


Figure 18: Density contours of  $p=1$  and  $p=3$  spatial-discretization schemes, at  $t=100$ , using IRK4 scheme and  $\Delta t = 0.05$





**Figure 19:** Comparison of the temporal accuracy as a function of time-step size for the BDF2 and IRK4 schemes at  $t = 0.1$  (a) for  $p=1$  spatial discretization, (b) for  $p=3$  spatial discretization.

Starting with the computed intermediate solution as the initial condition, Figures 17 and 18 depict the numerical results at  $t = 100$  for  $p = 1$  and  $p = 3$  spatial discretizations using the BDF2 and IRK4 time-integration schemes with the same fixed time-step size of  $\Delta t = 0.05$ , respectively. It can be observed that for a fixed temporal scheme (either the BDF2 or IRK4 scheme), the higher-order accurate  $p = 3$  spatial scheme provides the best shape-retaining convection capability: the vortices produced around the corners of the triangle wedge keep their shapes far downstream. On the other hand, the  $p = 1$  scheme is relatively dissipative as seen by the diffusion of the core of the vortices as they are convected downstream. While the improvement in the solution due to the increase in spatial discretization order is evident, any solution changes due to the use of higher-order temporal schemes are less evident in these qualitative illustrations.

In order to assess the accuracy of our temporal discretizations, we perform a temporal refinement study, similar to that described for the previous test case. The temporal error is isolated by producing a reference "exact" numerical solution, using a small time-step size of  $\Delta t = 2 \times 10^{-4}$  for each temporal and spatial discretization. Using the same initial and boundary conditions as described above, computations are performed using time-step sizes from 0.05 to  $5 \times 10^{-3}$  and the error is measured at  $t = 0.1$ . The results shown in Fig 19 indicate that the design accuracy of the BDF2 and IRK4 schemes is approached in both cases, yielding curve slopes of 1.92 and 3.96 in the case of the second-order accurate ( $p=1$ ) spatial discretization, and 1.72 and 3.96 in the case of the fourth-order accurate ( $p=3$ ) spatial discretization, for the BDF2 and IRK4 temporal schemes, respectively. However, it is notable that the absolute temporal errors of both temporal schemes are consistently smaller for the second-order accurate ( $p=1$ ) spatial discretization as compared to the higher-order spatial discretization case.

While the implicit scheme requires relatively few time steps compared to the explicit scheme, the cpu time required for a single explicit time step is much lower than that required for an implicit time step, since

**Table 4:** Explicit-implicit comparison for the shedding flow case

for solution at $t = 2.5$	time-step size	time steps	convergence limit	cpu time (s)
implicit (BDF2)	$\Delta t = 0.05$	50	$1 \times 10^{-7}$	5160
explicit (FD2)	$\Delta t = 5 \times 10^{-5}$	50000	—	22920

only a residual evaluation is required in the explicit case, as opposed to the solution of a large non-linear system of equations in the implicit case. Nevertheless, the results in Table 4 show that the implicit time-integration scheme requires 4.4 times less cpu time to integrate the solution to the non-dimensional time  $t = 2.5$  compared to the explicit approach in this case. However, this comparison is only qualitative in nature, since the temporal accuracy of the two approaches is not matched, and the cpu time required by the implicit approach is strongly dependent on the level of convergence required of the inner iterations for the implicit solver. However, the performance of the explicit scheme is determined by stability considerations, which will be detrimental when the resulting temporal accuracy imposed by the stability limit is not closely related to the desired temporal accuracy.

## 2.9 Extension to Diffusion Problems and the Navier-Stokes equations

The DG method has been initially developed for hyperbolic problems. Its use for problems involving diffusion is relatively recent, and was delayed by the lack of an obvious way to compute viscous interface fluxes between neighboring elements (i.e. the lack of a "viscous Riemann solver"). In this project, we have considered various approaches to extending our DG methodology to viscous flows, including the Navier-Stokes equations.

### 2.9.1 Unified Inviscid-Viscous Flux Formulation

We discuss here briefly (only in a one-dimensional framework) a methodology to devise stable interface conditions that reduces to the inviscid Riemann solve in the limit when the viscosity tends to zero. Such methodology has the advantage that it can be easily incorporated in existing DG codes. This research topic constitutes the basis for the PhD thesis of A. Rahunathan, a graduate student that was supported from this grant and who is expected to graduate in Fall 2008.

In order to study well-posedness, it is sufficient to consider a locally linearized, constant coefficient version of the full problem. The dependent variables appearing in the subsequent equations can therefore be considered as perturbations around a mean value, unless otherwise specified. Let us first consider the one-dimensional scalar linear advection-diffusion equation

$$\frac{\partial w}{\partial t} + a \frac{\partial w}{\partial x} = \epsilon \frac{\partial^2 w}{\partial x^2}, \quad x \in [-1, 1] \quad (34)$$

where  $a$  and  $\epsilon \geq 0$  are constants. Multiplying the equation with  $w$  and integrating in  $[-1, 1]$ , one can easily obtain

$$\begin{aligned} \frac{1}{2} \frac{d||w||_2^2}{dt} &= \int_{-1}^1 \epsilon w w_{xx} dx - \int_{-1}^1 a w w_x dx = \\ &= \epsilon \int_{-1}^1 \left[ \frac{d}{dx} (w w_x) - w_x^2 \right] dx - \int_{-1}^1 a w w_x dx = \\ &= \left( \epsilon w w_x - \frac{a}{2} w^2 \right)_{x=1} - \left( \epsilon w w_x - \frac{a}{2} w^2 \right)_{x=-1} - \epsilon ||w_x||_2^2 \end{aligned} \quad (35)$$



It follows that the PDE is well-posed if:

$$\left(\epsilon w w_x - \frac{a}{2} w^2\right)_{x=-1} \geq 0 \quad (36)$$

$$\left(\epsilon w w_x - \frac{a}{2} w^2\right)_{x=1} \leq 0 \quad (37)$$

For the purely hyperbolic case  $\epsilon = 0$ , these relations reduce to

$$\left(\frac{a}{2} w^2\right)_{x=-1} \leq 0 \quad (38)$$

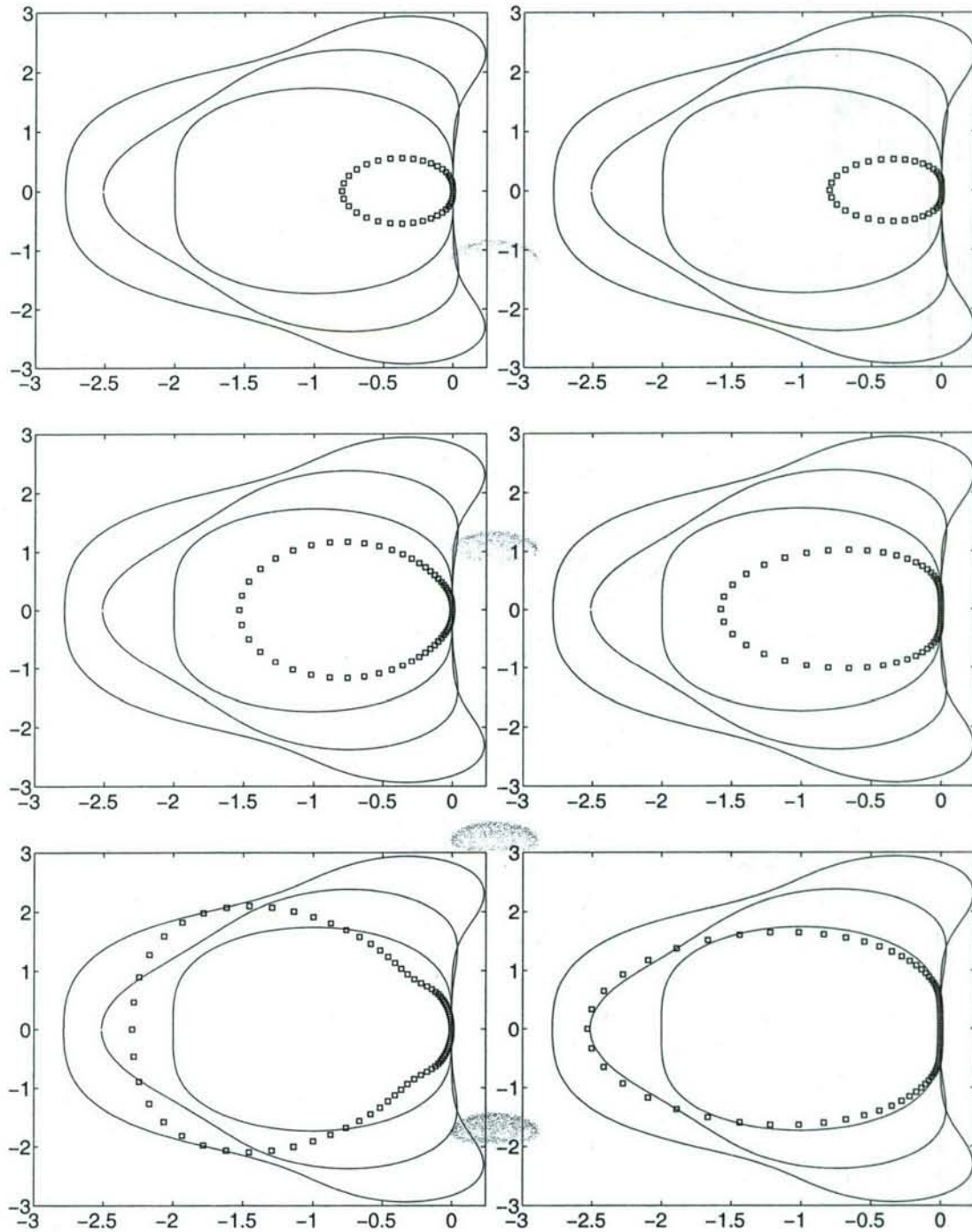
$$\left(\frac{a}{2} w^2\right)_{x=1} \geq 0 \quad (39)$$

and it follows that for  $a > 0$  one needs to set  $w = 0$  at  $x = -1$ , whereas for  $a < 0$  one needs  $w = 0$  at  $x = 1$ . Since  $w$  represents a perturbation to the mean value of the dependent variable, the meaning of these two conditions is that for the case  $a > 0$  one needs to specify the value of the dependent variable at  $x = -1$ , while for  $a < 0$  one needs to specify the value of the dependent variable at  $x = 1$ , as is well known for the linear advection equation. In order to develop boundary conditions for the parabolic case that reduce to the advection case when  $\epsilon = 0$ , let us first notice that the boundary term can also be written  $w(\epsilon w_x - a/2w)$ . Then one has the following two relevant cases: (1)  $a > 0$  Set  $w = 0$  for  $x = -1$ , and require  $(w\epsilon w_x - a/2w^2)_{x=1} < 0$ . A sufficient condition for the latter to hold is  $w_x = 0$  at  $x = -1$ . (2)  $a < 0$  Set  $w = 0$  for  $x = 1$ , and require  $(w\epsilon w_x - a/2w^2)_{x=-1} > 0$ . A sufficient condition for the latter to hold is  $w_x = 0$  at  $x = 1$ .

These boundary conditions can be seen to represent Dirichlet ( $w = 0$ ) and Neumann ( $w_x = 0$ ) type of boundary conditions, respectively, for the full dependent variable  $u$ , the perturbation of which is  $w$ . They can be easily incorporated in DG approximations of the linear advection-diffusion equation. Consider an interface between two elements, and let the value of the dependent variable computed from the element situated immediately left of the interface be denoted by  $u^L$ , the one from the right by  $u^R$ . The same superscripts are used to denote the two different values of the space derivative. It then follows that the interface values for the dependent variable and its derivative, which are denoted by a star superscript, should be computed as follows: (1)  $a > 0$ ,  $u^* = u^L$ ,  $u_x^* = u_x^R$ . (2)  $a < 0$ ,  $u^* = u^R$ ,  $u_x^* = u_x^L$ . Numerical results in figure 20 show the eigenvalues of the discrete operator obtained by the use of the above interface conditions versus the stability regions of the time discretization method.

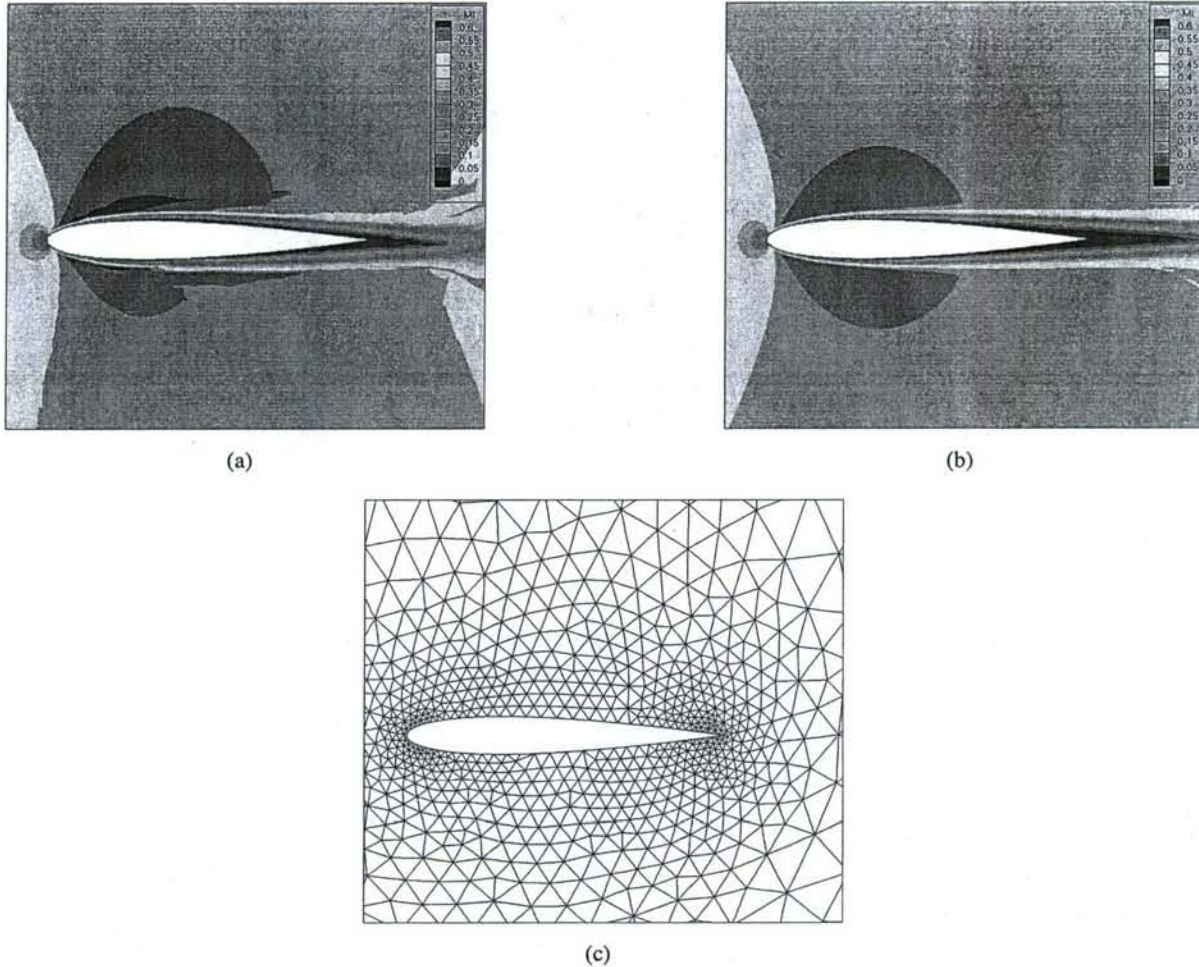
## 2.9.2 Interior Penalty Method

An alternate approach for extending purely hyperbolic DG discretizations to viscous flows is afforded by the interior penalty (IP) method. This method is among the simplest approach for discretizing diffusion problems in the context of a DG discretization and has a long history of development [70]. One of the desirable features of the IP method is that it retains the nearest neighbor stencil of the original DG discretization applied to hyperbolic problems. On the other hand, the IP approach involves the definition and use of a penalty parameter, which is required for stability. The choice of the penalty parameter has been mostly heuristic or empirical in most previous work, and this has been considered one of the principal drawbacks of this approach. However, recently, an explicit expression for the penalty parameter has been derived and validated [71], removing much of the empiricism of this approach. The two-dimensional steady-state h-p multigrid solver described previously has been extended to solve the Navier-Stokes equations using the IP method to discretize the viscous terms in the Navier-Stokes equations. This effort was undertaken near the



**Figure 20:** The eigenvalue spectra are plotted in the complex plane for the flux constructed by imposing stability. The solid lines show the stability regions of the classic Runge-Kutta schemes. The figures on the left show the eigen values spectrum for  $a = 1.0$  and  $\varepsilon = 10^{-2}$ ; The figures on the right show the eigen values spectrum for  $a = 1.0$  and  $\varepsilon = 10^{-3}$ . The figures from top to bottom show the numerical results obtained using 2,3 and 4 Gauss-Legendre points in the DG approximations, respectively.





**Figure 21:** Solution of viscous flow over NACA0012 airfoil at Mach=0.5, Reynolds number=5,000 and zero degrees incidence. (a) Solution using interior penalty method with  $p=1$  (second-order accurate). (b) Solution using interior penalty method with  $p=3$  (fourth-order accurate). (c) Triangular mesh containing 1,822 isotropic elements.

end of the project period of performance, and only preliminary results are available at this time. The laminar viscous flow over a NACA0012 airfoil was computed at a mach number of 0.5, and a Reynolds number of 5000 on a triangular mesh of 1,822 elements, which is depicted in Figure 21(a). The solution of the Navier-Stokes equations for this case, using a  $p=1$  (second-order accurate) discretization is shown in Figure 21(b), while the corresponding  $p=3$  (fourth-order accurate) solution on the same mesh is shown in Figure 21(c). While the second-order accurate solution displays non-smooth contours in the aft region of the airfoil, the fourth-order accurate solution is found to be very smooth. For this case, the flow is known to separate at 81% chord, from previous highly resolved finite-volume computations [72], and the  $p=3$  results are found to agree closely with this value. The smoothness and accuracy of these results is notable particularly in light of the coarseness of the employed mesh, as well as the use of isotropic elements throughout the domain, even in the presence of thin shear layers.

## 2.10 p-Multigrid for Nodal DG Approximation

Initial DG formulations [73] used as degrees of freedom the coefficients of the solution in a polynomial basis expansion (i.e. Legendre polynomials in one space dimension). This modal formulation seems more intuitive, but has a large drawback for nonlinear problems: to compute nonlinear fluxes, one needs to first compute the value of the solution by evaluating the polynomial expansion at the flux integration points, then evaluate the fluxes. Researchers realized later that this translation from modal space into physical space can be avoided through a collocation (nodal) formulation. In the latter framework the degrees of freedom are the values of the solution at the collocation points, so the flux values can be evaluated easily, in particular if the collocation points are chosen to coincide with the integration points [74, 75]. In the context of multigrid methods, the modal DG formulation again is more intuitive: the polynomial-type basis functions can be embedded hierarchically, and a  $p$ -multigrid approach is natural (the restriction operator for example just neglects extra modes); this is the approach taken in the work described in the previous sections. However, due to the advantage of the nodal approach for nonlinear problems, we investigated the feasibility of a  $p$ -type multigrid method in this latter formulation; such an investigation had not been reported previously in the literature. We considered the following initial-boundary value problem as our model problem:

$$\frac{\partial}{\partial t} \tilde{Q} + \nabla \cdot F = \tilde{S} \quad \text{in } \Omega \times \mathbb{R}^+, \quad (40)$$

$$Q|_{\partial\Omega \times \mathbb{R}^+} = g, \quad (41)$$

$$Q(x, t=0) = Q_0(x), \quad (42)$$

where  $\tilde{Q}$ ,  $F$  and  $\tilde{S}$  are the state, flux and source vectors, respectively and  $\Omega$  is a bounded spatial region in  $\mathbb{R}^d$ ,  $d = 1, 2, 3$ . Subdividing the domain of interest into non-overlapping elements to enable the accurate resolution of the approximate solution curve with a minimum number of elements, and mapping each element individually onto the master element  $[-1, 1]$  using a linear transformation, the equation (40) becomes

$$\frac{\partial}{\partial t} Q + \frac{\partial}{\partial \xi} F = S \quad (43)$$

where the new variables are the transformed components of the state and source vector

$$Q = J\tilde{Q}, \quad S = J\tilde{S}$$

and  $J = dx/d\xi$ . A spectral approximation is constructed by use of the basis of Lagrange interpolants

$$\phi_i = \prod_{j=0, j \neq i}^n \frac{\xi - \xi_j}{\xi_i - \xi_j}, \quad \phi_i(\xi_j) = \delta_{ij} \quad (44)$$

through the  $n+1$  Gauss-Legendre quadrature points  $\xi_i, i = 0, 1, \dots, n$ . The discontinuous Galerkin approximation (see, for example, [74]) approximates the solution, flux and source by polynomials in a trial space  $P_n$ , i.e.

$$Q(\xi, t) = \sum_{i=0}^n q_i(t) \phi_i(\xi) \quad (45)$$

where the nodal values of the flux and source are computed from the nodal values of the solution,  $q_i(t)$ , at the Gauss-Legendre quadrature point  $\xi_i$  and time  $t$ , i.e.

$$f_i(t) = F(Q(\xi_i, t)) = F(q_i(t)), \quad s_i(t) = S(Q(\xi_i, t)) = S(q_i(t)).$$



Doing a Galerkin projection and using discrete orthogonality relations between Legendre polynomials, the final discrete form of the equations governing each variable at the Legendre-Gauss points can be written as

$$\frac{d}{dt}q_j = N(q_j) \quad (46)$$

where  $j = 0, 1, \dots, n$  and

$$N(q_j) = 1/\omega_j \int_{-1}^1 F \frac{d\phi_j}{d\xi} d\xi - 1/\omega_j [F\phi_j]_{-1}^1 + 1/\omega_j \int_{-1}^1 S\phi_j d\xi \quad (47)$$

$$= \sum_{k=0}^n f_k \omega_k / \omega_j \phi'_j(\xi_k) - 1/\omega_j [F\phi_j]_{-1}^1 + s_j. \quad (48)$$

In  $p$ -multigrid, higher order ( $p$ ) and lower order ( $p-1$ ) interpolants serve as the fine and coarse grids, respectively. Consider the solution of the discrete problem on the fine grid

$$\frac{d}{dt}u^p = N(u^p) \quad (49)$$

where  $u^p$  is the discrete solution vector for  $p^{th}$  order interpolation on a given grid. The current approximation of the solution  $u^p$  is denoted as  $\bar{u}^p$ , which is obtained by approximately solving (49) by using the nodal discontinuous Galerkin method. Since  $\bar{u}^p$  does not satisfy the above equation exactly, its substitution into equation (49) yields

$$N(\bar{u}^p) - \frac{d}{dt}\bar{u}^p = r^p$$

where  $r^p$  is the discrete residual. For the solution of steady-state problems, we are only concerned with the final state where all time derivatives vanish. Thus the discrete residual becomes

$$r^p = N(\bar{u}^p). \quad (50)$$

The main difficulty resides in defining the restriction and prolongation operators. We discuss this briefly here, and refer to [76] for full details. Since the polynomial approximation spaces are nested, we can write the  $i^{th}$  basis function of order  $p-1$ ,  $\phi_i^{p-1}$ , in terms of the basis functions of higher order  $p$ ,  $\{\phi_j^p\}$ :

$$\phi_i^{p-1} = \sum_j \phi_i^{p-1}(\xi_j^p) \phi_j^p. \quad (51)$$

To form the residual restriction operator, we return to the definition of the residual vector from (47), (50):

$$r_j^p = 1/\omega_j^p \int_{-1}^1 F \frac{d\phi_j^p}{d\xi} d\xi - 1/\omega_j^p [F\phi_j^p]_{-1}^1 + 1/\omega_j^p \int_{-1}^1 S\phi_j^p d\xi. \quad (52)$$

Given  $r^p$  we need to determine  $r^{p-1} = \tilde{I}_p^{p-1} r^p$ . Writing out  $r_i^{p-1}$  like (52) and using (51) yields

$$r_i^{p-1} = \sum_j (\phi_i^{p-1}(\xi_j^p) \omega_j^p / \omega_i^{p-1}) r_j^p.$$

Thus, the residual restriction operator is

$$\begin{aligned} \tilde{I}_p^{p-1} &= \alpha^{p-1} \\ \alpha_{i,j}^{p-1} &= \phi_i^{p-1}(\xi_j^p) \omega_j^p / \omega_i^{p-1}. \end{aligned} \quad (53)$$

The prolongation operator,  $I_{p-1}^p$ , transfers  $u^{p-1}$  to the next finer level. Thus, we seek a representation of the coarse level solution on the finer level [58]. That is, we wish to calculate  $u^p$  such that

$$\sum_j u_j^p \phi_j^p = \sum_i u_i^{p-1} \phi_i^{p-1} \quad (54)$$

$$u_j^p = (I_{p-1}^p)_{j,i} u_i^{p-1}. \quad (55)$$

Using (51) and (54) we have

$$\begin{aligned} \sum_j u_j^p \phi_j^p &= \sum_i u_i^{p-1} \sum_j \phi_i^{p-1}(\xi_j^p) \phi_j^p \\ &= \sum_j \sum_i \phi_i^{p-1}(\xi_j^p) u_i^{p-1} \phi_j^p. \end{aligned} \quad (56)$$

Since a state representation is unique in the basis  $\phi_j^p$ , from (55) and (56) it must be the case that

$$(I_{p-1}^p)_{j,i} = \phi_i^{p-1}(\xi_j^p).$$

Numerical tests with these transfer operators indicate that for some problems the nodal DG method may offer advantages over the modal DG method. Results for a solution of the Euler equations are reported in Table 5 and Figure 22.

**Table 5:** nodal DG vs. modal DG for the fixed order  $p = 4$

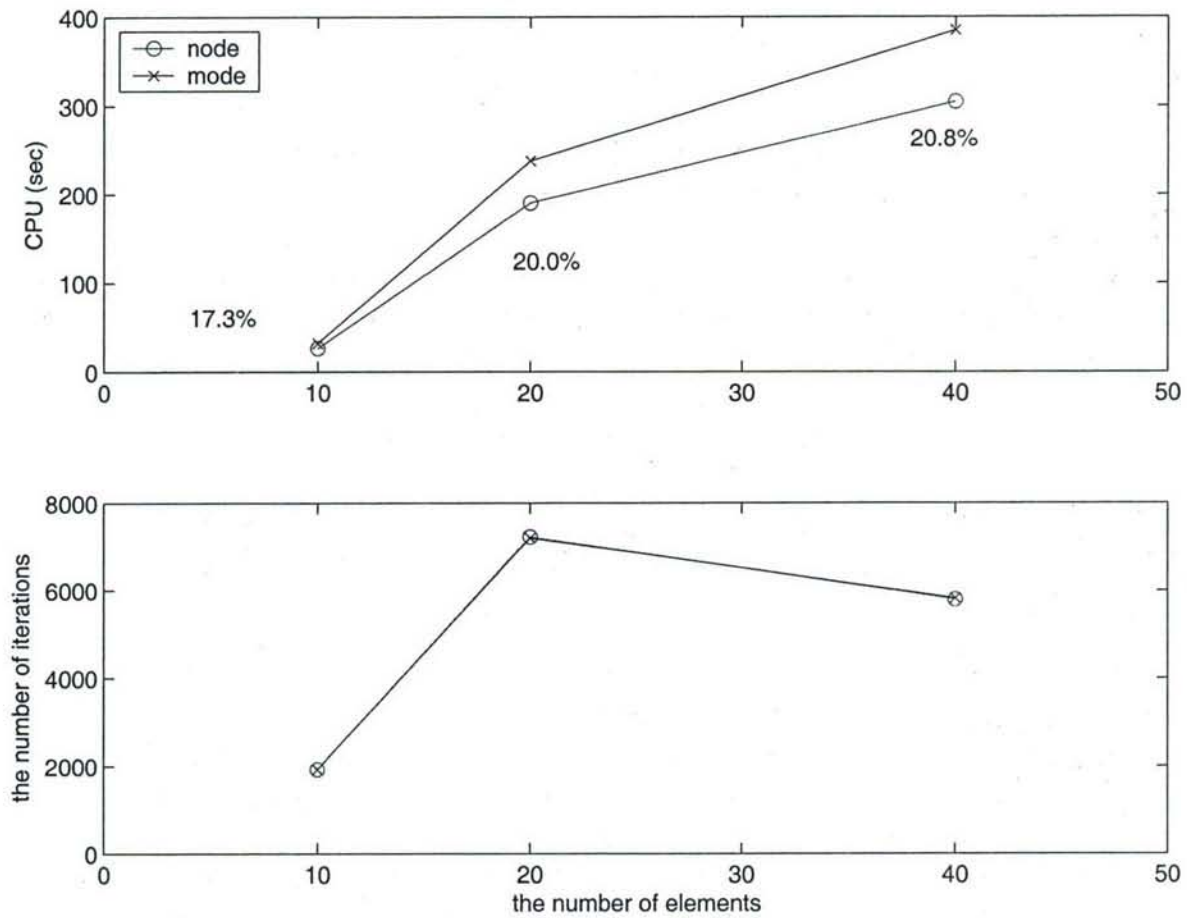
num of elements	CPU time (sec)			num of iterations		
	10	20	40	10	20	40
nodal DG	26.57	190.36	304.85	1925	7231	5794
saving	17.3%	20.0%	20.8%	.	.	.
modal DG	32.14	238.01	385.01	1927	7202	5821

### 2.11 Adjoint-Based Error Estimation and Adaptivity

As mentioned in the introduction, one of the advantages of the use of unstructured meshes is the possibility of easily performing adaptive mesh refinement for improved accuracy. For higher-order methods, the order of the discretization may also be modified adaptively and locally in order to improve accuracy. The combination of mesh refinement (h-refinement) and changing the order of accuracy of the discretization (p-refinement), known as h-p adaptivity, has been shown to lead to asymptotically optimal error reduction in the literature [31]. However, one of the principal obstacles towards effective use of all adaptive methods lies in the ability to formulate effective and reliable error estimation procedures for driving the adaptive refinement process.

Rather than attempting to quantify the discretization error at every point in the mesh for adaptive refinement criteria, an alternative approach involving adjoint-based error prediction methods has been gaining popularity recently [33–35]. In this approach, solution of the adjoint of the governing flow equations is used to quantify the sensitivity of an objective function with respect to local grid resolution throughout the domain. This approach to mesh refinement is non-local, in the respect that it provides for the global effect of local resolution changes. Objective functions representing engineering quantities of interest such as lift, drag, or  $L/D$ , can be constructed, thus enabling the mesh refinement process to target quantities of interest. Since vastly different grid resolution distributions are required for different engineering applications of





**Figure 22:** Nodal DG vs. modal DG in CPU time (top) and the number of iterations (bottom) when the number of elements varies for the fixed order  $p = 4$ . The percentages are savings in CPU time for nodal DG.

the same problem, this approach enables accurate and efficient engineering results by avoiding excessive resolution in areas of little influence on the quantity of interest.

In order to estimate the error in an output functional such as drag or lift, the adjoint solution for this quantity must first be computed. The change in the output functional which would occur if the mesh was uniformly refined can then be estimated by interpolating the flow solution computed on the current mesh to a fully refined mesh, and computing the residual on this refined mesh. Because the interpolated solution does not correspond to the more accurate solution which would be obtained by solving the discretized equations on the finer mesh, the equations on the finer mesh are not satisfied by the interpolated solution and a non-zero residual field is obtained. The error in the output functional (i.e. the change in the functional which would be observed if the flow was solved on the fully refined mesh) is then given as the inner product between the non-zero residual field and the adjoint solution field [77, 78]. Furthermore, the contribution from each cell in the mesh to this inner product can be taken as an estimate of the local error contribution to the output functional. Thus, in order to adaptively reduce the error in the output functional, a suitable mesh refinement strategy is one which flags cells with large relative contributions to the inner product error estimate for refinement.

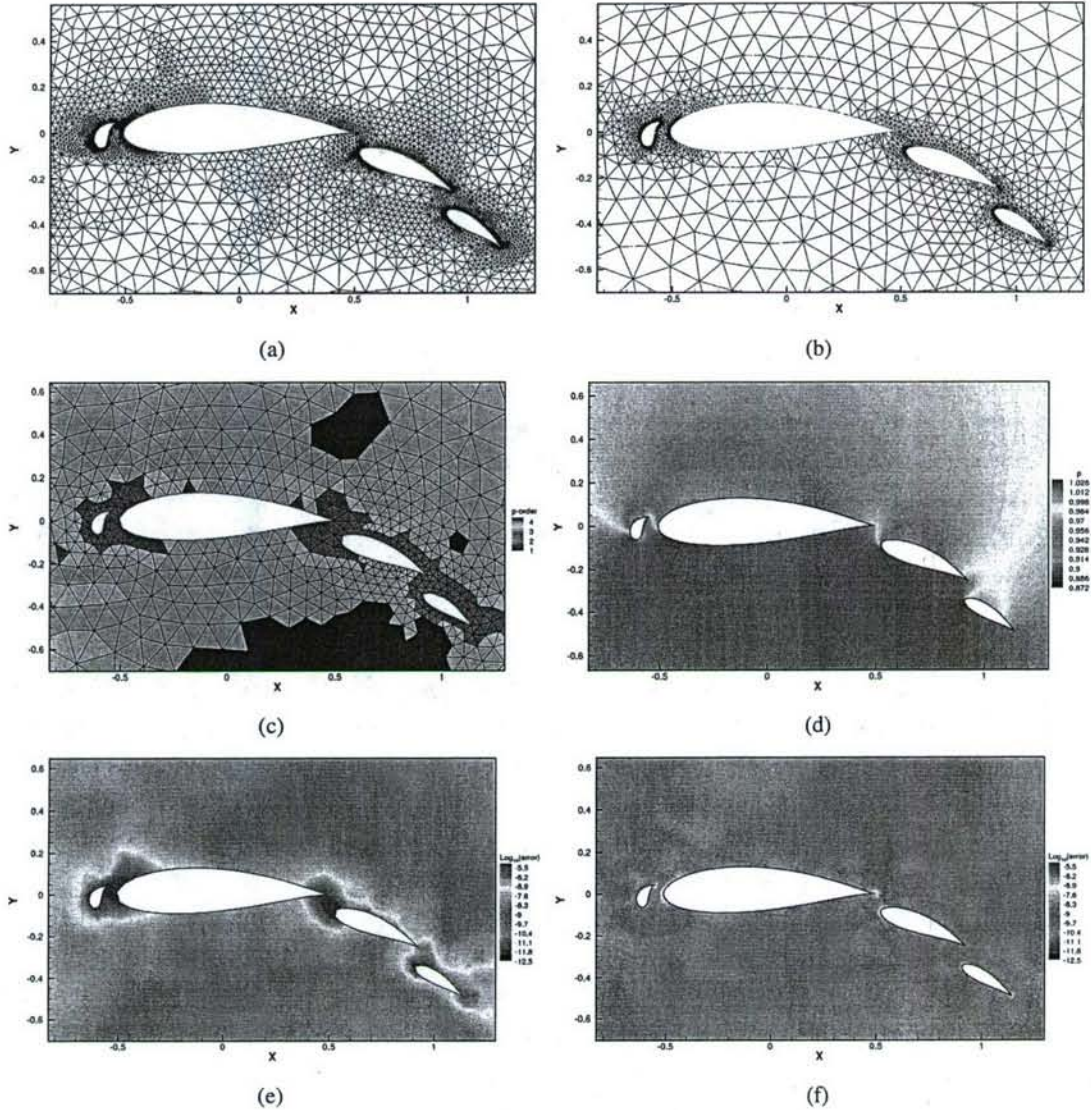
The adjoint error estimation procedure described above is obviously intended for driving an h-refinement procedure. However, a similar adjoint error estimation technique can be used to drive a p-refinement procedure. In this approach, a p-order solution is first computed, and then a p+1 order solution is reconstructed using gradient reconstruction techniques [78]. When this reconstructed solution is substituted into the p+1 discretization equations, non-zero residuals are obtained and the error estimate is given as the inner product between the adjoint solution and non-zero residual vector.

In the current project, both h-refinement and p-refinement adjoint-based error criteria have been implemented. The adjoint equations themselves are obtained by linearizing the discrete equations of the analysis problem and then transposing the resulting Jacobian matrices. Because the many of the terms in the full Jacobian are already computed for the implicit element Jacobi scheme used in the solver, this task was relatively straight-forward. Once the adjoint equations are assembled, the solution of the adjoint problem is carried out using the h-p multigrid approach, following the strategy used to solve the flow equations, and resulting in similar convergence rates for the flow and adjoint problems.

Figure 23 illustrates both h- and p-refinement procedures for an inviscid flow test case in two dimensions. The configuration consists of an idealized four element airfoil geometry, which results in strong local gradients in the flow field and illustrates well the advantages of unstructured meshes for such configurations. The Mach number for this case is 0.2, and the incidence is 0 degrees. The initial mesh contains 1,590 elements, and was originally run with a p=0 discretization. The targeted output functional is the drag coefficient, which ideally should be zero for this two-dimensional inviscid flow. The adjoint solution for the drag coefficient is computed and used to drive the adaptation procedure in all cases. The refined mesh obtained after three refinement passes is shown in Figure 23(a), using a fixed p=0 discretization, while the h-refined mesh using a fixed p=2 discretization is shown in Figure 23(b). As expected, much of the refinement occurs near the airfoil surfaces, particularly near the leading edges, and the amount of refinement is substantially reduced for the p=2 discretization. Figure 23(c) depicts the p-refinement distribution obtained for the same problem using a fixed mesh resolution, but variable p-order discretization throughout the domain, as determined by the adjoint error estimation technique. While much of the domain contains relatively low (p=0, p=1) order discretizations, the critical areas near the airfoil surfaces and leading edges are seen to employ up to p=3 discretizations. Figure 23(d) illustrates the computed solution for the p-adaptive test case, showing



good resolution of all flow features on this relatively coarse grid. The distribution of error in the drag coefficient as predicted by the adjoint procedure is shown in Figures 23(e) and (f), for the initial calculation ( $p=0$ ) and the final  $p$ -adapted calculation, illustrating the spatial distribution of error and its reduction through the adaptive process.



**Figure 23:**  $h$ - and  $p$ -adaptive mesh refinement for low-speed inviscid flow over idealized four element airfoil geometry using adjoint error estimation based on drag coefficient. (a) Adaptively  $h$ -refined mesh for constant  $p=0$  discretization. (b) Adaptively  $h$ -refined mesh for constant  $p=2$  discretization. (c) Adaptive  $p$ -distribution for  $p$ -refinement scheme using fixed mesh resolution ( $h$ ). (d) Computed Mach contours for  $p$ -adaptive case. (e) Error distribution computed with adjoint formulation for output drag coefficient on initial mesh and discretization. (f) Error distribution computed with adjoint formulation for output drag coefficient on final  $p$ -adaptive discretization.

While the incorporation of adjoint methods for driving  $h$  and  $p$  adaptive refinement techniques has been pursued in this project, techniques for optimally combining  $h$  and  $p$  refinement remain under investigation.



### 3. Conclusions and Further Work

The work performed under this project has significantly advanced our understanding of solution techniques for high-order discretizations suitable for industrial flow problems. On the one hand, we have shown that the h-p multigrid approach delivers optimal convergence rates, which are both insensitive to the order of accuracy  $p$  of the discretization, and of the level  $h$  of mesh resolution, even for relatively large problems in three dimensions. Additionally, this approach has been shown to scale very well on current day parallel computers, using up to 2000 processors. On the other hand, robustness issues were encountered particularly for the larger three-dimensional problems using the higher order accurate discretizations, and these issues remain to be investigated. Resolution of these effects may likely require the use of limiters for locally reducing the order of accuracy of the discretization in non-smooth regions, or the addition of artificial diffusion, or the use of filtering. While the performance of the h-p multigrid solver is near optimal in terms of convergence per cycle, the cost of the element Jacobi smoother used within each cycle grows significantly with increasing order of accuracy  $p$ , and the development of less computationally intensive smoothers which maintain these optimal convergence rates remains an active area of research.

In this project, we have also demonstrated the extension of the h-p multigrid approach to viscous flow problems, unsteady problems, and for nodal DG formulations. In order to develop a production level high-order accurate solver, the current techniques must all be extended into three dimensions, and suitable turbulence models or sub-grid scale models must be implemented for turbulent flow simulations. The work performed in this project is currently being leveraged in an on-going NASA funded project for high-order accurate methods for high-speed flows, and will form the basis for a three-dimensional Navier-Stokes DG solver.

The adjoint error estimation and adaptivity techniques developed in this project hold great promise for improving the accuracy and reliability of engineering simulations where a small number of output objectives are of interest. The adjoint approach has been used to drive h- and p-adaptive refinement techniques individually, although the combination of h-p refinement is still under investigation. This remains a non-trivial task, since the decision to switch between h-refinement or p-refinement rests not on the local error magnitude, but on the local smoothness of the solution, which is not addressed by the adjoint approach itself. Thus, additional techniques for examining solution smoothness and for choosing between h and p refinement must be developed and demonstrated. This is particularly important for flows with discontinuities such as shock waves or slip lines.

### 4. Broader Impact of DEPSCoR Project

As a component of the DEPSCoR program, the objectives of this project include not only the technical accomplishments described above, but also the broader impact in building research infrastructure and capabilities at the University of Wyoming and across the state. As mentioned in the original proposal, the University of Wyoming (UW) has chosen to concentrate on several selective areas of distinction, one of which is computational science. The PI and co-PI of this project were hired with this emphasis in mind, and were newcomers to the University of Wyoming at the initiation of this project. Indeed this project constituted the first major research award for the PI and co-PI at UW, and has been instrumental in jump-starting their research programs. At the end of this project, the co-PI is successfully involved in two other computational science collaboration research projects, while the PI has built up a fully externally funded research group



of 3 post-doctoral researchers and four PhD graduate students. Furthermore, much of the on-going research in these programs has been leveraging progress made in the current ONR DEPSCoR project. For example, the code developed by the PI's research group is being used as the basis for a high-order DG simulation research project for high-speed flows funded by the NASA Fundamental Aeronautics program.

The current award also initiated the fruitful collaboration between the Departments of Mathematics and Mechanical Engineering which has continued and grown over the last three years. Campus wide, the PI and co-PI have been instrumental in leading in the organization of a computational science and engineering program, designed for providing graduate students with better interdisciplinary training in computational science matters.

Finally, in the fall of 2006, a partnership between the State of Wyoming, the University of Wyoming and the National Center for Atmospheric Research (NCAR) in Boulder Colorado was announced, whereby a Petaflops computational facility will be hosted within the State of Wyoming and close collaboration between NCAR and UW in computational science matters will be pursued. While the funding provided by the State, and the negotiations between the University and NCAR executives played a decisive role in the formation of this partnership, the roots of this collaboration can be traced partly to the current ONR DEPSCoR project. Our work under this project on DG methods has been very similar in nature to research being performed at NCAR for the next generation climate simulation code, and grass-roots technical collaborations developed over the course of this project between the PI and co-PI and NCAR scientists. It was through this collaboration that the efforts of NCAR in establishing a new Petaflops facility were communicated, which lead to initial contacts between NCAR executives and the UW administration, facilitated by the PI and co-PI of this project.

#### 4.1 People Supported by the Grant

- Dimitri Mavriplis (PI), Professor, Department of Mechanical Engineering
- Dan Stanesco (co-PI), Assistant Professor, Department of Mathematics
- Cris Nastase, Postdoctoral Researcher, Department of Mechanical Engineering
- Li Wang, PhD Student, Department of Mechanical Engineering
- Arunasalam Rahunathan, PhD Student, Department of Mathematics
- Khosro Shabazi, Postdoctoral Researcher, Department of Mechanical Engineering (March 2007 - May 2007)

#### 4.2 Publications Resulting from this Grant

##### Journal Publications

1. L. Wang and D. J. Mavriplis, *Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations*, Journal of Computational Physics, Vol 225, No. 2, August 2007, pp 1994-2015.

2. C. Nastase and D. J. Mavriplis, *High-Order Discontinuous Galerkin Methods using an hp Multigrid Approach*, Journal of Computational Physics, Vol 213, No.1, March 2006, pp 330-357.
3. A. Rahunanthan and D. Stanescu; *Well-Posed Interface Conditions for Discontinuous Galerkin Approximations of the Navier-Stokes Equations*, to be submitted to Journal of Scientific Computing.

#### Conference Papers

1. C. Nastase and D. J. Mavriplis, *A Parallel h-p Multigrid Solver for Three Dimensional Discontinuous Galerkin Discretizations of the Euler Equations*, AIAA paper 2007-0060, paper presented at the AIAA Aerospace Sciences Meeting and Exhibit, Reno NV, January 2007.
2. L. Wang and D. J. Mavriplis, *Implicit Solution of the Unsteady Euler Equations for High-Order Accurate Discontinuous Galerkin Discretizations*, AIAA paper 2006-0109, paper presented at the AIAA Aerospace Sciences Meeting and Exhibit, Reno NV, January 2006.
3. C. Nastase and D. J. Mavriplis, *Discontinuous Galerkin Methods Using an h-p Multigrid Solver for Inviscid Compressible Flows on Three-Dimensional Unstructured Meshes*, AIAA paper 2006-0107, paper presented at the AIAA Aerospace Sciences Meeting and Exhibit, Reno NV, January 2006.
4. C. Nastase and D. J. Mavriplis, "High-Order Discontinuous Galerkin Methods using a Spectral Multigrid Approach", AIAA paper 2005-1268, paper presented at the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno NV, January 2005.
5. D. Kim and D. Stanescu; "*p*"-Multigrid For The Nodal Discontinuous Galerkin Approximation, Proceedings of the 9th Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, April 2-7, 2006.

#### Conference Presentations w/o Papers

1. D. J. Mavriplis, *Solution of high-order discontinuous Galerkin methods using a combined H-P multigrid approach*, presentation at the 9th Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, April 2-7, 2006.
2. C. Nastase and D. J. Mavriplis, *High-Order Discontinuous Galerkin Methods using an hp Multigrid Approach*, presentation at the SIAM Annual Meeting, New Orleans, LA, July 2005.
3. L. Wang and D. J. Mavriplis, *Implicit Solution of High-Order Accurate Discretizations of the Unsteady Wave Equation using Spectral Multigrid*, presentation at the 12th Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, April 2005.

#### Student Theses

1. L. Wang; *Techniques for Discontinuous Galerkin Discretizations*, PhD Thesis, in preparation, 2008.
2. A. Rahunanthan; *Stable Discontinuous Galerkin Approximations for Parabolic Partial Differential Equations*, PhD Thesis in preparation, 2008.



### References

- [1] P. R. Spalart, W-H. Jou, M. Strelets, and S. R. Allmaras. Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach. Paper presented at the First AFOSR International Conference on DNS and LES, Louisiana Tech University, Ruston, Louisiana, August 1997.
- [2] K. Sreenivas, D. Hyams, B. Mitchell, L. Taylor, D. Marcum, and D. Whitfield. Computation of vortex-intensive incompressible flow fields. AIAA Paper 2002-3306, June 2002.
- [3] S. Murman and M. Aftosmis. Cartesian grid simulations of a canard-controlled missile with a spinning tail. AIAA Paper 2003-3670, June 2003.
- [4] R. S. Hirsch. Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique. *Journal of Computational Physics*, 19:90–109, 1975.
- [5] Y. Adam. Highly accurate compact implicit methods and boundary conditions. *Journal of Computational Physics*, 24:10–22, 1977.
- [6] P. Collela and P. Woodward. The piecewise-parabolic (ppm) method for gas-dynamical simulations. *Journal of Computational Physics*, 54:174–201, 1984.
- [7] J. F. James. *A student's guide to Fourier transforms with applications in physics and engineering*. Cambridge University Press, 1995.
- [8] E. Cuthill and J. McKee. Reducing the band width of sparse symmetric matrices. In *Proc. ACM Nat. Conference*, pages 157–172, 1969.
- [9] A. Pothen, H. D. Simon, and K. P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11:430–452, 1990.
- [10] V. Venkatakrishnan and D. J. Mavriplis. Implicit solvers for unstructured meshes. *Journal of Computational Physics*, 105(1):83–91, June 1993.
- [11] T. J. Barth. Parallel CFD algorithms on unstructured meshes. In *Special Course on Parallel Computing in CFD*, pages 7–1, 7–41, May 1995. AGARD Report-807.
- [12] D. J. Mavriplis. *Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes*. PhD thesis, Princeton University, MAE Department, 1987.
- [13] J. Peraire, J. Peirö, and K. Morgan. A 3D finite-element multigrid solver for the Euler equations. AIAA Paper 92-0449, January 1992.
- [14] M. Lallemand, H. Steve, and A. Dervieux. Unstructured multigriding by volume agglomeration: Current status. *Computers and Fluids*, 21(3):397–433, 1992.
- [15] D. J. Mavriplis and V. Venkatakrishnan. A unified multigrid solver for the Navier-Stokes equations on mixed element meshes. *International Journal for Computational Fluid Dynamics*, 8:247–263, 1997.
- [16] D. J. Mavriplis. On convergence acceleration techniques for unstructured meshes. AIAA paper 98-2966, presented at the 29th AIAA Fluid Dynamics Conference, Albuquerque, NM, June 1998.
- [17] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate Essentially Non-Oscillatory schemes III. *J. Comp. Phys.*, 71:231–303, 1987.
- [18] R. J. DiPerna. Finite difference schemes for conservation laws. *Comm. Pure Appl. Math.*, 25:379–450, 1982.

- [19] T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the euler and navier-stokes equations. Technical report, AGARD R787, 1992.
- [20] T. J. R. Hughes. Recent progress in the development and understanding of SUPG methods with special reference to the euler and navier-stokes equations. *Int. J. Num. Meth. Fl.*, 7:1261–1275, 1987.
- [21] C. Johnson. Finite element methods for flow problems. Technical report, AGARD R787, 1992.
- [22] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. ii. general framework. *Math. Comput.*, 52(186):411–435, April 1989.
- [23] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. iii. one-dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.
- [24] J. N. Reddy. *An Introduction to the Finite Element Method, 2nd Edition*. McGraw Hill, 1993.
- [25] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Laboratory Report LA-UR-73-479, 1973.
- [26] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. iv. the multidimensional case. *Math. Comput.*, 54(190):545–581, 1990.
- [27] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. v. multidimensional systems. *Journal of Computational Physics*, 141:199–221, 1998.
- [28] G. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1978.
- [29] P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:375–372, 1981.
- [30] G. Jiang and C.-W. Shu. On cell entropy inequality for discontinuous Galerkin methods. *Math. Comput.*, 62:531–538, 1994.
- [31] P. Develoo, J. T. Oden, and P. Pattani. An h-p adaptive finite-element method for the numerical simulation of compressible flows. *Comp. Meth. Appl. Mech. Eng.*, 70:203–235, 1988.
- [32] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral methods in fluid dynamics*. Springer-Verlag, New York, NY, 1988.
- [33] J. D. Mueller and M. B. Giles. Solution adaptive mesh refinement using adjoint error analysis. AIAA Paper 2001-2550, June 2001.
- [34] D. A. Venditti and D. L. Darmofal. A grid adaptive methodology for functional outputs of compressible flow simulations. AIAA Paper 2001-2659, June 2001.
- [35] M. Park. Adjoint-based, three-dimensional error prediction and grid adaptation. AIAA Paper 2002-3286, June 2002.
- [36] A. Jameson, J. J. Alonso, J. J. Reuther, L. Martinelli, and J. C. Vassberg. Aerodynamic shape optimization techniques based on control theory. AIAA Paper 98-2538, June 1998.
- [37] W. K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. AIAA paper 97-0643, January 1997.



- [38] M. Duta, M. Giles, and L. Lafronza. Sensitivity analysis for aeroacoustic applications. AIAA-Paper 2003-3204, June 2003.
- [39] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138:251–285, 1997.
- [40] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35(6):2440–2463, 1998.
- [41] T. C. Warburton, I. Lomtev, Y. Du, S. J. Sherwin, and G. E. Karniadakis. Galerkin and discontinuous Galerkin spectral/hp methods. *Comput. Methods Appl. Mech. Engrg.*, 175:343–359, 1999.
- [42] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *SIAM J. Sci. Comput.*, 16(3):173–261, 2001.
- [43] S. F. Davis. Simplified second-order Godunov-type methods. *SIAM J. Sci. Statist. Comput.*, 9(3):445–473, 1988.
- [44] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.
- [45] A. Harten, P. D. Lax, and B. Van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.
- [46] F. Eleuterio Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Applied Mechanics. Springer-Verlag, New York, NY, 1999.
- [47] P. Batten, N. Clarke, C. Lambert, and D. M. Causon. On the choice of wavespeeds for the HLLC Riemann solver. *SIAM J. Sci. Comput.*, 18(2):1553–1570, 1997.
- [48] P. Batten, M. A. Leschiner, and U. C. Goldberg. Average-state jacobians and implicit methods for compressible viscous and turbulent flows. *J. Comput. Phys.*, 137:38–78, 1997.
- [49] Pavel Solin, P. Segeth, and I.D. Zel. *High-Order Finite Element Methods*. Studies in Advanced Mathematics. Chapman and Hall, 2003.
- [50] Barna Szabo and Ivo Babuska. *Finite Element Analysis*. John Wiley & Sons, Inc., New York, NY, 1991.
- [51] S. Pirzadeh. Three-dimensional unstructured viscous grids by the advancing-layers method. *AIAA Journal*, 34(1):43–49, 1996.
- [52] S. Pirzadeh. Private Communication, 2006.
- [53] D. A. Dunavant. Highr degree efficient symmetrical gaussian quadrature rules for the triangle. *Int. J. Numer. Meth. Engng.*, 21:1129–1148, 1985.
- [54] D. A. Dunavant. Economical symmetrical quadrature rules for complete polynomials over a square domain. *Int. J. Numer. Meth. Engng.*, 21:1777–1784, 1985.
- [55] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comput.*, 54(545), 1990.
- [56] U. Trottenberg, A. Schuller, and C. Oosterlee. *Multigrid*. Academic Press, London, UK, 2000.
- [57] D. J. Mavriplis. An assessment of linear versus non-linear multigrid methods for unstructured mesh solvers. *Journal of Computational Physics*, 175:302–325, January 2002.

- [58] Krzysztof J. Fidkowski and David L. Darmofal. Development of a higher-order solver for aerodynamic applications. In *Proceedings of the 42nd Aerospace Sciences Meeting and Exhibit, Reno NV, 2004*. AIAA Paper 2004-0436.
- [59] D. J. Mavriplis. Multigrid techniques for unstructured meshes. In *VKI Lecture Series VKI-LS 1995-02*, March 1995.
- [60] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message Passing Interface*. MIT Press, Cambridge, MA, 1994.
- [61] George Karypis. Metis, university of minnesota, department of computer science. <http://www-users.cs.umn.edu/karypis/metis>, 2003.
- [62] R. Biswas, M. J. Djomehri, R. Hood, H. Jin, C. Kiris, and S. Saini. An application-based performance characterization of the Columbia Supercluster. Paper presented at the 2005 Supercomputing Conference, Seattle, WA., November 2005.
- [63] E. Hairer and G. Warner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, 1991.
- [64] C. A. Kennedy and M. H. Carpenter. Additive Runge-Kutta schemes for convection diffusion reaction equations. *Applied Numerical Mathematics*, 44(1):139–181, January 2003.
- [65] Hester Bijl, Mark H. Carpenter, Veer N. Vatsa, and Christopher A. Kennedy. Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: Laminar flow. *J. Comput. Phys.*, 179:313–329, 2002.
- [66] G. Jothiprasad, D. J. Mavriplis, and D. A. Caughey. Higher order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes. *J. Comput. Phys.*, 191(2):542–566, Nov. 2003.
- [67] F. Davoudzadeh, H. McDonald, and B. E. Thompson. Accuracy evaluation of unsteady CFD numerical schemes by vortex preservation. *Computers and Fluids*, 24:883–895, 1995.
- [68] E. Garnier, P. Sagaut, and M. Deville. A class of explicit ENO filters with application to unsteady flows. *J. Comput. Phys.*, 170:184–204, 2000.
- [69] H. C. Yee, N. D. Sandham, and M. J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *J. Comput. Phys.*, 150:199–238, 1999.
- [70] B. Cockburn, G. E. Karniadakis, and C. W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer, 2000.
- [71] K. Shahbazi. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics*, 205(2):401–407, 2005.
- [72] D. J. Mavriplis, A. Jameson, and L. Martinelli. Multigrid solution of the Navier-Stokes equations on triangular meshes. AIAA Paper 89-0120, January 1989.
- [73] B. Cockburn and C.-W. Shu. Tvb runge-kutta local projection discontinuous Galerkin finite element method for conservation laws ii: general framework. *Math. Comp.*, 52:411–435, 1989.
- [74] S.L. Woodruff D.A. Kopriva and M.Y. Hussaini. Discontinuous spectral element approximation of Maxwell's equations. *Lecture notes in Computational Science and Engineering*, 11:355–361, 2001. Springer, Berlin.
- [75] M.Y. Hussaini D. Stanescu and F. Farassat. Aircraft engine noise scattering by fuselage and wings: a computational approach. *Journal of Sound and Vibration*, 263:319–333, 2003.



- [76] D. Kim and D. Stanescu. " $p$ "-multigrid for the nodal discontinuous Galerkin approximation. Proceedings of the 9th Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, April 2006.
- [77] D.A. Venditti and D.L. Darmofal. Grid adaptation for functional outputs: Application to 2-D inviscid compressible flow. *Journal of Computational Physics*, 176:40–69, 2002.
- [78] P. Houston, R. Rannacher, and E. Suli. A posteriori error analysis for stabilized finite-element approximations of transport problems. *Comput. Methods Appl. Mech. Engr.*, 190:1483–1508, 2000.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE (DD-MM-YYYY) 31-08-2007		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 01-June-2004 to 31-May-2007	
4. TITLE AND SUBTITLE Efficient High-Order Accurate Methods using Unstructured Grids for Hydrodynamics and Acoustics				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER N00014-04-1-0602	
				5c. PROGRAM ELEMENT NUMBER	
				5d. PROJECT NUMBER	
6. AUTHOR(S) Mavriplis, Dimitri, J. Stanescu, Dan				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Wyoming 1000E University Ave. Laramie WY 82071				8. PERFORMING ORGANIZATION REPORT NUMBER DODONR44500EXT:: DODONR44500USL	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research Ballston Centre Tower One 800 North Quincey Street Arlington, VA 22217-5660				10. SPONSOR/MONITOR'S ACRONYM(S) ONR	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution is Unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The objective of this project has been the development of high-order accurate simulation techniques for fluid flow problems of interest to the US Navy, such as hydrodynamics and acoustics. Efficient solution techniques for high-order Discontinuous Galerkin methods have been investigated from both a theoretical and practical standpoint. An h-p multigrid solution strategy which delivers optimal convergence rates which are independent of both the order of accuracy p of the discretization, and the resolution h of the mesh has been developed and demonstrated on steady and unsteady problems, showing good efficiency and parallel scalability using up to 2000 processors. Sensitivity analysis techniques based on the solution of the adjoint problem have also been developed, and used to drive h and p adaptive refinement techniques for increasing accuracy at optimal cost. Future work will concentrate on extending these techniques to three dimensional Reynolds-averaged Navier-Stokes (RANS) simulations and large eddy simulations (LES) for important flows of Naval interest.					
15. SUBJECT TERMS Computational Fluid Dynamics Acoustics Hydrodynamics					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  ul	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dimitri Mavriplis
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			19b. TELEPHONE NUMBER (Include area code) 307-766-2868